(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-515001
(P2004-515001A)
(43)公表日　平成16年5月20日(2004.5.20)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 12/14	G06F 12/14	320B	5B017
H03K 19/173	H03K 19/173	101	5J042
H04L 9/10	H04L 9/00	621A	5J104

審査請求 未請求　予備審査請求 有　（全 90 頁）

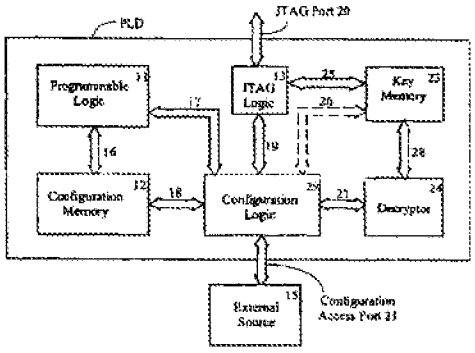
(21)出願番号　特願2002-546976(P2002-546976)
(86)(22)出願日　平成13年11月28日(2001.11.28)
(85)翻訳文提出日　平成15年5月15日(2003.5.15)
(86)国際出願番号　PCT/US2001/045056
(87)国際公開番号　WO2002/044876
(87)国際公開日　平成14年6月6日(2002.6.6)
(31)優先権主張番号　09/724,652
(32)優先日　平成12年11月28日(2000.11.28)
(33)優先権主張国　米国(US)
(81)指定国　EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),CA,JP

(71)出願人　591025439
ザイリンクス　インコーポレイテッド
ＸＩＬＩＮＸ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国　カリフォルニア州　９５１２４－３４００　サン　ホセ　ロジック　ドライブ　２１００
(74)代理人　100064746
弁理士　深見　久郎
(74)代理人　100085132
弁理士　森田　俊雄
(74)代理人　100083703
弁理士　仲村　義平
(74)代理人　100096781
弁理士　堀井　豊

最終頁に続く

(54)【発明の名称】解読アルゴリズムおよび解読鍵を有するプログラマブルロジックデバイス

(57)【要約】
プログラマブルロジックデバイス（ＰＬＤ）に実現される設計の複製を防ぐために、ＰＬＤ自体が、設計者がロードした解読鍵を記憶し、暗号化された構成ビットストリームがＰＬＤにロードされる際にこれを解読するための解読器を含む。ＰＬＤはまた、ビットストリームが暗号化されているかどうかを示すヘッダ情報を読出すためのロジックを含み、暗号化されたビットストリームも暗号化されないビットストリームも受付けることができる。暗号化鍵は、不揮発性メモリに記憶されてもよく、または電源が抜かれたときに保持されるように電池でバックアップされてもよい。



【特許請求の範囲】

【請求項１】

暗号化されたビットストリームを解読するための解読器と、解読器によって用いられる鍵とを有するＰＬＤにおいて、ＰＬＤを用いる方法であって、

ＰＬＤを非安全モードにするステップと、

鍵をＰＬＤにロードするステップとを含む、方法。

【請求項２】

鍵をロードするステップの後にＰＬＤを安全モードにするステップをさらに含む、請求項１に記載のＰＬＤを使用する方法。

【請求項３】

構成メモリにより構成される構成可能ロジックと、

ＰＬＤの外部源からビットストリームを受けるための構造と、

解読鍵を記憶するための鍵メモリと、

鍵を用いてビットストリーム中の暗号化された構成ビットを解読することにより構成データを形成するための解読アルゴリズムを有する解読器と、

構成データを構成メモリにロードするための構造とを含む、プログラマブルロジックデバイス（ＰＬＤ）。

【請求項４】

構成データを構成メモリにロードするための構造は、ＣＲＣチェックサム計算回路を含む、請求項３に記載のＰＬＤ。

【請求項５】

構成メモリから構成をリードバックするための構造と、

ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すとき構成をリードバックするための構造を不能化するための構造とをさらに含む、請求項３に記載のＰＬＤ。

【請求項６】

ＰＬＤが構成された後ＰＬＤを再構成するための構造と、

ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すときＰＬＤを再構成するための構造を不能化するための構造とをさらに含む、請求項３に

記載のＰＬＤ。

【請求項７】

解読器は、複数個の解読鍵を記憶するためのレジスタの１つから、解読のために別の鍵も使用されるかどうかを示す値を読出す、請求項３に記載のＰＬＤ。

【請求項８】

解読器は、鍵によって特定される態様と異なったように鍵を用いようとするならば、解読を中止するための回路を含む、請求項３に記載のＰＬＤ。

【請求項９】

鍵は、鍵の組のうち最初の鍵か、中間の鍵か、最後の鍵か、または唯一の鍵かを特定する、請求項３に記載のＰＬＤ。

【発明の詳細な説明】

【０００１】

［発明の分野］

この発明はＰＬＤに関し、より特定的にはビットストリームによりＰＬＤにロードされた設計の保護に関する。

【０００２】

［発明の背景］

ＰＬＤ（プログラマブルロジックデバイス）は、設計者によって選択されるデジタル論理機能を行なう集積回路構造である。ＰＬＤは、論理ブロックおよび配線を含み、典型的には論理ブロックも配線もプログラム可能である。ＰＬＤの一般的なタイプの１つはＦＰＧＡ（フィールドプログラマブルロジックデバイス）で、論理ブロックが、典型的にはルックアップテーブルおよびフリップフロップを含み、典型的にはそれらの入力信号のいかなる機能も生成し記憶することができる。別のタイプはＣＰＬＤ（コンプレックスプログラマブルロジックデバイス）で、論理ブロックがＡＮＤ機能およびＯＲ機能を行ない、入力信号の選択がプログラム可能である。

【０００３】

ＰＬＤの外部のビットストリームを記憶することの問題

ＰＬＤにおいて実現される設計は複雑になってきており、設計を完成しデバックしＰＬＤ内で実現するのに何ヶ月もかかることがしばしばある。設計が、ＰＬＤが一部であるシステムとなり、利潤を得るために販売される場合、設計者はこの設計努力の結果が他人に複製されることを望まない。設計者はその設計をトレードシークレットにすることをしばしば望む。多くのＰＬＤ、特にＦＰＧＡは、ＰＬＤに電源が投入されるたびに、ＰＲＯＭなどの外部装置からロードされなければならない揮発性構成メモリを使用する。構成データはＰＬＤの外部に記憶され構成アクセスポートを介して送信されなければならないので、たとえば基板パターン上にプローブを置くことで、構成アクセスポート上のデータをモニタする攻撃者により設計のプライバシーはたやすく侵害されかねない。

【０００４】

現在の解決策およびその欠点

設計を暗号化する努力がなされてきたが、設計を攻撃者から安全にしかつ合法的ユーザによって使用しやすくすることは困難である。暗号化アルゴリズムは問題ではない。いくつかの暗号化アルゴリズム、たとえば、標準データ暗号化規格（ＤＥＳ）およびより安全な高度暗号化規格（ＡＥＳ）アルゴリズムが、データのブロックを暗号化するものとして公知である。暗号化されていないデータワードと次の暗号化されたデータワードとの排他的論理和を取りその後に解読をする暗号ブロック連鎖（ＣＢＣ）のプロセスにより、ＤＥＳまたはＡＥＳは、データの直列ストリームを暗号化することができ、したがってこれらはビットストリームを暗号化してＰＬＤを構成するのに適切である。設計を暗号化するために使用される鍵は、ＰＬＤと設計を解読する構造との間で安全な態様で何らかの形でやり取りされなければならないので、その設計は、ＰＬＤを構成するために使用される前にＰＬＤによって解読され得る。そして、暗号化されていない設計を用いて一旦ＰＬＤが構成されてしまうと、設計は引続き無許可の発見から保護されなければならない。

【０００５】

ザイリンクス・インコーポレイテッド（Ｘｉｌｉｎｘ，　Ｉｎｃ．）のＰｅｔｅｒ　Ａｌｆｋｅによる、「構成の問題：電源投入、揮発性、安全性、電池バックアップ」（“Ｃｏｎｆｉｇｕｒａｔｉｏｎ　Ｉｓｓｕｅｓ：　Ｐｏｗｅｒ－ｕｐ，　Ｖｏｌａｔｉｌｉｔｙ，　Ｓｅｃｕｒｉｔｙ，　Ｂａｔｔｅｒｙ　Ｂａｃｋ－ｕｐ”）と題する１９９７年１１月２４日付の刊行物は、ＦＰＧＡ内に設計を保護するための特定のアーキテクチャ的特徴を持たない既存のＦＰＧＡデバイス内の設計を保護するために取り得るいくつかのステップを記載する。１つの方法は、設計構成データをＦＰＧＡにロードし、次に構成データのソースを除去するが、電池を用いてＦＰＧＡへ連続的な電源を維持しながらＦＰＧＡを待機非動作モードに保持することである。しかしながら、電池に対する電源要求のため、この方法は大規模なＦＰＧＡデバイスには実践的でない。

【０００６】

別の可能性は、不揮発性構成メモリである。デバイスを販売する前に設計を工場

でロードする場合、構成されたＰＬＤデバイスの購入者が、設計が何であるかを判断することは難しい。しかしながら、リバースエンジニアリング工程により、すなわち、プログラムされたデバイスの覆いを取除き、金属層を除去し、不揮発性メモリセルを化学的に処理することで、どのメモリセルが充電されているのかが暴露され、そのため攻撃者は設計を知ることができてしまう。さらに、不揮発性メモリは、標準のＣＭＯＳプロセス技術よりもより複雑でより高価なプロセス技術を必要とし、市場に投入するのに時間がかかる。

【０００７】

解読鍵をＰＬＤの不揮発性メモリに記憶し、暗号化されたビットストリームをＰＬＤにロードし、ＰＬＤ内の鍵を用いてそのビットストリームを解読することも公知である。これにより、攻撃者は、ＰＬＤにロードされているときにビットストリームを読出すことができず、ＰＬＤから電源が引抜かれたとき鍵を保持できる。そのような構成は、米国特許第５，３８８，１５７５号においてＡｕｓｔｉｎにより記載される。しかし、この構造は、ユーザの設計を攻撃のすべてのモードから保護しない。

【０００８】

設計保護に加えて、ユーザの中にはデータ保護を必要とする者もいる。彼らは、ＰＬＤが電源を失うとき失われるべきでないデータをＰＬＤ内に生成したかもしれない。そのようなデータを保護することが望まれる。

【０００９】

便利で確実かつ安全な設計保護方法が依然必要とされている。

［発明の概要］

この発明は、無許可の使用およびデータ損失からＰＬＤを保護するためのいくつかの構造および方法を提供する。

【００１０】

電源投入時にロードされなければならないスタティックＲＡＭメモリによってＰＬＤが構成される場合、構成データは、デバイスにロードされているときに保護されなければならない。先行技術におけるように、これは、構成データを暗号化しそれを集積回路装置の外部のメモリに記憶し、１つ以上の解読鍵をＰＬＤにロ

ードし、電源が切られたときＰＬＤにその鍵を維持し、鍵を用いて構成データを解読する解読回路をＰＬＤ内に含め、ＰＬＤ内で解読された構成データを生成し、解読された構成データを用いてＰＬＤを構成することにより達成される。

【００１１】

追加のセキュリティのため、不揮発性メモリを用いて鍵を保存するのではなく、この発明は好ましくは、ＰＬＤに接続される電池を用いて、電源がＰＬＤから引抜かれたとき鍵を保存する。鍵を不揮発性メモリに記憶するＰＬＤを取外し、ＰＬＤの覆いを取除き、不揮発性ビットのうちどれが論理１にプログラムされどれが論理０にプログラムされているのかを観察することが可能であるが、スタティックメモリセルにのみ記憶された鍵の内容を判断することは非常に困難であると考えられる、というのも鍵を記憶するためにさえも、鍵を記憶するメモリセルの電源が維持されなければならず、動作電源がＰＬＤに対して連続している状態で、ＰＬＤの覆いを取除き層を除去しプローブを行なわなければならないはずだからである。

【００１２】

ＰＬＤにロードされた設計を攻撃者が盗み得る態様

鍵が十分なセキュリティを提供しなければ、攻撃者は暗号化コードを解読し、鍵の値を判定してしまうであろう。周知のデータ暗号化規格ＤＥＳは５６ビットの暗号化鍵を用いており、高度なコンピュータで数時間のうちに解読されてしまい鍵を暴露してしまう。ＤＥＳは、Ｊｏｈｎ Ｗｉｌｅｙ ＆ Ｓｏｎｓ， Ｉｎｃ．により発行された「応用暗号手法第二版：プロトコル、アルゴリズム、およびＣ言語でのソースコード」（“Ａｐｐｌｉｅｄ Ｃｒｙｐｔｏｇｒａｐｈｙ Ｓｅｃｏｎｄ Ｅｄｉｔｉｏｎ：ｐｒｏｔｏｃｏｌｓ，ａｌｇｏｒｉｔｈｍｓ，ａｎｄ ｓｏｕｒｃｅ ｃｏｄｅ ｉｎ Ｃ”）著作権１９９６年の２６５～２７８頁においてＢｒｕｃｅ Ｓｃｈｎｅｉｅｒにより記載される。そのような周知の暗号化規格を用いることが望ましい場合、セキュリティを増強するために、構成データは、毎回異なった鍵を用いて何回も暗号化され得り、暗号化が繰返されるたびに約2^{56}だけ暗号化コードを強化する。もしくは、それは、第１の鍵を用いて暗号化され、第２の鍵を用いて解読され、第３の鍵を用いて暗号化されても

よく、三重ＤＥＳ規格の一部である組合せであってもよい。他の暗号化アルゴリズムを用いてもよく、セキュリティは鍵にあるのでアルゴリズムを秘密にする必要はない。暗号化方法が対称的であるとき、暗号化に使用されるのと同じ鍵をＰＬＤに記憶し逆の順で用いて解読を行なう。

【００１３】

複数の鍵を提供するＰＬＤにおいて、使用されるべき鍵の数およびすべての鍵のアドレスが、暗号化されていないビットストリームで与えられるならば、攻撃者は、一度に１つ鍵を攻撃して鍵の値をより簡単に判断できてしまうであろう。そのような攻撃を回避するために、いくつの鍵が使用されのか、および鍵がある組のうち最後の鍵であるのかまたはさらに以下に続くのかの表示を、ビットストリームではなく鍵の中に記憶することにより、さらなるセキュリティが達成される。

【００１４】

ビットストリームがＰＬＤにロードされた後ビットストリームをリードバックするというオプションをＰＬＤが提供する場合、攻撃者が用い得る別の方法は、このビットストリームをリードバックすることである。設計を攻撃するこの方法を回避するために、ある実施例では、リードバックを提供しかつ暗号化も提供するＰＬＤは、暗号化が使用されてしまうとリードバック特徴を不能化する能力を含む。別の実施例では、リードバックする能力を提供するＰＬＤは、リードバックされる前に構成データを暗号化する。

【００１５】

さらに、いくつかのＰＬＤは、部分的構成（設計のいくつかの部分をロードするためにいくつかの構成アドレスが特定される）、および部分的再構成（既存の設計が消去されてから新しい設計データがロードされる）というオプションを提供する。ＰＬＤがこれらのオプションを提供する場合、攻撃者は、ＰＬＤを部分的に再構成し、設計の連続部分を見えるようにして、おそらくは設計全体を知ることが可能である。そのような攻撃を回避するために、ある実施例では、暗号化された設計がロードされたＰＬＤの部分的構成および再構成は不能化される。別の実施例では、いくつかの構成アドレスが特定され得るが、アドレスは暗号化され

る。

【００１６】

攻撃のさらに別のモードは、ＰＬＤのセキュリティ状態を示すビットを反転しようとすることである。動作電圧を低下または上昇させ、温度を変化させ、ノイズをある種のポートに与えることが想定される。そのようなビット反転から保護するために、ＰＬＤが暗号化されたビットストリームで作動しているとき、安全モードフラグがセットされ、ある実施例では、このフラグが設定解除されると、すべての構成データが消去される。デバイスが作動している間でも再構成を許可しない別の実施例では、構成データは、ビットストリームが送信される前に消去される。

【００１７】

攻撃の別のモードは、暗号化されたビットストリームの一部を再配置し、それらが非暗号化されるとき、設計者が意図しないＰＬＤの見える部分に置かれるようにすることである。この再配置を防ぐために、暗号化および解読プロセスにおいてアドレス情報を用いて、暗号化されたビットストリームの一部を、設計者が意図したものと異なるＰＬＤ場所に送信することにより、それが意味を有さないデータに異なって解読される。暗号ブロック連鎖（ＣＢＣ）は、この結果を達成する効果的な手段の１つである。暗号ブロック連作において、解読されたデータパケット（ブロック）は、排他的論理和関数を用いて次のデータブロックと組合され、その後に次のブロックを解読し、そのため各データブロックごとの暗号化されたデータは、それに先行するあらゆるブロックおよびこれらのブロックの順番に依存する。データの同一のブロックは、それらに先行するデータブロックの値に依存して異なった値に暗号化される。このように、もしブロックの順序が変更されるならばビットストリームは正しく解読されない、というのも暗号化されたビットストリームが再配置される場所が後続のデータにスクランブルをかけるからである。さらに、初期ＣＢＣ値を修正してデータのアドレスを組込み解読されたデータが特定の位置に置かれるようにして正しく解読を行なうことができる。

【００１８】

代替的に、ＰＬＤが、設計の一部が暗号化され一部が暗号化されないことを可能

にしているならば、攻撃者は、設計の暗号化された部分についての情報を読出そうとする暗号化されていない部分を暗号化された部分に加えるかもしれない。したがって、設計が完全に暗号化されるかまたは完全に暗号化されないことを許可するが、これらが混ざらないようにすることにより、さらなるセキュリティが達成される。さらにまた、ある実施例では、データが暗号化されているとき、構成データの単一の開始アドレスの後に単一のフルチップ構成のみを許可することにより、追加のセキュリティが提供される。

【００１９】

さらに、便利な検査およびデバックを可能にし、かつＰＬＤ製造者がその顧客（設計を製造しＰＬＤを構成する設計者）と自由にやり取りすることを可能とするために、ＰＬＤは、動作の暗号化モードおよび非暗号化モードの両方を有し、暗号化モードで動作するとき、ＰＬＤへの構成データのロードを制御する構成ビットストリームの一部はまだ暗号化されない。

【００２０】

攻撃の別のモードとして、ＰＬＤ製造業者が、構成データをロードするためのヘッダ情報およびアドレスを含む、構成ビットストリームフォーマットについての情報を自由に与え、使用された暗号化方法についての情報を与える場合、この周知の情報を暗号化することで、暗号化鍵が暴かれるかもしれない。そのような暴露は、実際の構成データのみを暗号化し制御情報を非暗号化のままにすることにより回避される。

【００２１】

ＰＬＤ製造業者が、鍵メモリが安全モードおよび非安全モードの両方で使用されることを可能にしている場合、攻撃者は、鍵メモリを非安全モードにし鍵を読出すことにより、鍵を簡単に知ることができるかもしれない。そのような攻撃を回避するため、ＰＬＤ製造業者は、鍵メモリが非安全モードに移されると、ＰＬＤにロードされたすべての鍵およびいずれの構成データも消去されるようにする回路を含める。

【００２２】

［詳細な説明］

図１は、ＦＰＧＡ１０の先行技術の構造を示す。ＦＰＧＡはプログラマブルロジック１１を含み、プログラマブルロジック１１は典型的には、（１）ルックアップテーブル組合せ論理関数生成器と、ルックアップテーブルの出力および他の値を記憶するためのフリップフロップと、プログラマブルロジックの論理能力を向上させるためのマルチプレクサおよび論理ゲートとを有する論理ブロックと、（２）信号をＦＰＧＡのまわりに経路付けるための経路およびプログラマブル相互接続点と、（３）経路とＦＰＧＡの外部ピンとの間で信号を駆動するための入力／出力ブロックとを含む。

【００２３】

ＦＰＧＡはまた、経路選択トランジスタをオンにし、マルチプレクサを制御し、ルックアップテーブルを記憶し、入力／出力ブロックを制御するための構成メモリ１２を含み、このすべては、ＦＰＧＡを構成し設計者の所望する機能を行なうするためのものである。バス１６は、構成メモリ１２をプログラマブルロジック１１に接続し、典型的には、ＦＰＧＡ全体にわたっている制御線の分布された組である。いくつかのザイリンクス・インコーポレイテッドの製品（たとえばＸＣ６２００）は、プログラマブルロジック１１により構成ロジック１４がプログラミング情報を構成メモリ１２に送るバス１７を含んでいる。そのような構造は、米国特許第５，７０５，９３８号においてＫｅａｎによって記載される。

【００２４】

ＦＰＧＡ１０はさらに、ＪＴＡＧポート２０とインターフェイスし、特にＦＰＧＡが置かれる基板の検査を意図したＪＴＡＧ論理ブロック１３を含む。ＪＴＡＧ論理ブロック１３は、ＩＥＥＥ規格１１４９．１のスーパーセットであるＩＥＥＥ規格１５３２を実現する。ＪＴＡＧは基板レベルでの設計のデバッグを可能にする。

【００２５】

最後に、ＦＰＧＡ１０は、構成アクセスポート２１で外部源１５からの構成ビットストリームに応答し、ＪＴＡＧ論理ブロック１３とインターフェイスする構成ロジック１４を含む。構成アクセスポート２１上のビットストリームは、ワード、ある実施例では３２ビットのワードとして扱われる。通常ビットストリーム

の開始または開始付近でのワードのいくつかは、構成プロセスを設定するために用いられ、たとえば構成メモリフレームの長さ、および構成データの開始アドレスを含む。バス１９は、構成ロジック１４とＪＴＡＧ論理ブロック１３との通信を可能にし、ＪＴＡＧポートを別の構成アクセスポートとして用いることができる。バス１８は、構成論理ブロック１４と構成メモリ１２との間の通信を可能にする。特に、それは、メモリ１２の構成フレームを選択するためのアドレスと、書込動作および読出動作を行なうための制御信号と、構成メモリ１２にロードするかまたは構成メモリ１２からリードバックするデータとを運ぶ。

【００２６】

構成論理ブロック１４は、命令およびデータを受け、命令に従ってデータを処理する。これらの命令は、ビットストリームとして構成ロジック１４に入る。命令、またはヘッダの後には通常、処理すべきデータが続く。図２ａは、例示のビットストリーム構造を示す。ヘッダＡは処理を特定し、単一のワード、すなわちデータＡが後続することを特定する。ヘッダＢは処理を特定し、この場合には堀すべき４ワードのデータが後続することを特定する。

【００２７】

図２ｂは、ザイリンクス・インコーポレイテッドから入手可能なバーテックス（Ｖｉｒｔｅｘ）（Ｒ）において使用されるビットストリームにおける３２ビットのヘッダワードのデフォルトのフォーマット（フォーマットタイプ００１）である（バーテックス（Ｒ）は、この発明の譲受人であるザイリンクス・インコーポレイテッドの登録商標である）。このフォーマットは、フォーマットタイプ（００１）を示す３ビットと、演算コードを特定するための２ビットと、構成論理レジスタアドレスのための１６ビットと、ワードカウントのための１１ビットとを含む。演算コードは、読出動作、書込動作、または演算なしを指定することができる。たとえば、００は演算なしを指定することができ、０１は読出を指定することができ、１０は書込を指定することができる。ワードカウントのための１１ビットは、2^{11}ワードまたは２０４８ワードを特定することができる。図２ｃに示すように、ワードカウントがこれよりも大きければ、フォーマットタイプ００１でのワードカウントビットは００００００００００００に設定され、フォ

ーマットタイプ００１のヘッダの後にはフォーマットタイプ２のヘッダが続く。フォーマットタイプ２は２７ビットを用いてワードカウントを特定し、したがって2^{27}ワードまたは２６８万ワードを特定することができる。

【００２８】

図２ｄは、バーテックス（Ｒ）ビットストリームのヘッダが構成ロジック１４のレジスタにロードすることのできる制御情報の種類を示す。たとえば、構成論理レジスタアドレス００００を有するヘッダ（フォーマット００１）は、次の３２ビットのデータワードが巡回冗長検査（ＣＲＣ）レジスタにロードされるべきことを明示する。（バーテックス（Ｒ）のデバイスは、１６ビットの巡回冗長検査値を用いるので、いくつかのビットは０で埋められる。）ヘッダがアドレス０００１を含む場合、構成メモリ１２中のフレーム（列）を明示してデータを受けたり与えたりするために、次のデータがフレームアドレスレジスタにロードされる。

【００２９】

図２ｂに示す構成論理レジスタアドレス（１６ビット）は、図２ｄの左側の列に示す４ビットの値を与え、この４ビットの値は、次の３２ビットのデータワードを置くべき構成ロジック１４（図１）中のレジスタの１つを選択する。フレーム長レジスタ（アドレス１０１１）は、構成データがロードされるフレームの長さを特定する。（フレーム長、または列高さは、ＰＬＤの寸法に依存する。より大規模なＰＬＤは通常より高い列またはより長いフレームを有する。ＰＬＤに異なった構造を設けてデータワードをフレームに置くのではなく、ビットストリームのフレーム長を特定しフレーム長をレジスタに記憶することで、内部の構成ロジックは異なった寸法のＰＬＤに対して同一のものとなる。）

リードバックに関しては、読出コマンドが演算コードフィールドに置かれ、フレームデータ出力レジスタがアドレスされ、その後にワードカウントが続く（必要であればコマンドヘッダフォーマット２を用いる）。特定の数のワードが、フレームアドレスレジスタに特定されるアドレスから始まって構成メモリ１２からリードバックされ、構成アクセスポート２１またはＪＴＡＧポート２０のいずれかでシフトアウトされる。（リードバックデータは、リードバック命令を発行した

ポートに返される）。

【００３０】

ビットストリームヘッダまたはヘッダの対にワードカウントを特定することで（図２ｂおよび図２ｃ）、データワードがロードされるときカウントダウンするカウンタを設定する。多くの構成論理レジスタアドレスについて、ワードカウントは１である。ビットストリームヘッダが、構成データがロードまたはリードバックされていることを示す００１０または００１１の構成論理アドレスを有する場合、ワードカウントはもっと大きくなる。これは、図２ｃのヘッダフォーマット２が使用された場合である。フレームデータ入力レジスタ（アドレス００１０）を介して構成メモリ１２にロードされたデータ、またはフレームデータ出力レジスタ（アドレス００１１）を介して読出されたデータは、設計データと呼ばれる、というのもこれは、ＦＰＧＡに設計を実現させたり設計の状態を示したりするからである。他のレジスタデータは制御データである、というのもそれらは、ロジックが構成またはリードバックされている間に構成ロジックがどのように振る舞うかを制御するからである。

【００３１】

バーテックス（Ｒ）デバイスの構成についての一層の詳細は、９５１２４　カリフォルニア州、サン・ノゼ、ロジック・ドライブ、２１００のザイリンクス・インコーポレイテッド（この発明の譲受人）により２０００年１０月９日に出版された「バーテックス構成ガイド」（“Ｖｉｒｔｅｘ　Ｃｏｎｆｉｇｕｒａｔｉｏｎ　Ｇｕｉｄｅ”）に見出すことができる。

【００３２】

構成ロジック１４は典型的には、入来する構成ビットストリームに対して巡回冗長検査を行ない（Ｅｒｉｃｋｓｏｎの米国特許第５，３２１，７０４号または上に参照された「バーテックス構成ガイド」の３９から４０ページを参照されたい）、構成されている部分のフレーム長および構成データのワードカウントを示すヘッダビットを読出し、構成データをどこにロードすべきかを識別するアドレス情報を読出し、構成データのフレームを収集してそれらをアドレスに示される構成メモリ１２の列にロードする。構成ロジック１４はまた、構成メモリ１２か

ら外部の場所への構成データおよびフリップフロップ値のリードバックを制御する。ザイリンクス・インコーポレイテッドから入手可能なバーテックス（Ｒ）ＦＰＧＡにおいては、リードバックは、ＪＴＡＧポート２０または構成アクセスポート２１のいずれかを介して行なうことができる。

【００３３】

構成ロジック１４はまた、プログラマブルロジック１１から構成データを受けることができる。ＦＰＧＡの一部がＦＰＧＡの別の部分を構成する先行技術のＦＰＧＡ構造についてのさらなる情報は、Ｋｅａｎの米国特許第５，７０５，９３８号に見られる。バーテックス（Ｒ）アーキテクチャと同様のＦＰＧＡのアーキテクチャについての一層多くの情報は、Ｙｏｕｎｇらの米国特許第５，９１４，６１６号に見出せる。この発明の譲受人であるザイリンクス・インコーポレイテッドから入手可能なバーテックス（Ｒ）製品で用いられるビットストリームのフォーマットは、９５１２４　カリフォルニア州、サン・ノゼ、ロジック・ドライブ、２１００のザイリンクス・インコーポレイテッドから入手可能で２０００年１０月４日に発行された「バーテックスＦＰＧＡシリーズの構成およびリードバック」（“Ｖｉｒｔｅｘ ＦＰＧＡ Ｓｅｒｉｅｓ Ｃｏｎｆｉｇｕｒａｔｉｏｎ ａｎｄ Ｒｅａｄｂａｃｋ”）と題するアプリケーションノートＸＡＰＰ１３８に記載される。

【００３４】

解読付きのＰＬＤ

図３は、この発明のある実施例に従うＦＰＧＡ（ＰＬＤの１タイプ）のブロック図を示す。いくつかの要素は図１に示すのと同じであり、同じ参照番号が付与されており、再び説明しない。さらに、図３は、拡張構成論理ユニット２９と、解読器２４と、鍵メモリ２３とを含む。図３は、鍵メモリ２３がバス２５でＪＴＡＧアクセスポート２０からロードされる実施例を示す。他の実施例では、鍵メモリ２３は、別のポートを介してロードされる。バス２５は、書込および読出動作を行なうためのデータ、アドレスおよび制御信号を運び、ＪＴＡＧポート２０からの解読鍵のプログラミングを可能にする。ある実施例では、バス２６は、構成ポートからの鍵のプログラミングを可能にする。別の実施例では、バス２６が削

除される。さらに別の実施例では、バス２６が存在しバス２５が削除される。ここにさらに記載する実施例では、バス２６は、鍵メモリ２３から構成ロジック２９へセキュリティデータを運ぶ。ある実施例では、バス２７は、構成ロジック２９から解読器２４へ暗号化された構成データを運び、解読された構成データを構成ロジック２９に戻す。バス２８は、解読器２４が鍵にアクセスしてデータを解読することを可能にする。図３の構造に暗号化されたデータがロードされているとき、ビットストリームがロードされているのをモニタする攻撃者は、暗号化されたビットストリームのみを受信し、この方法によりユーザの設計を知ることができない。

【００３５】

部分的に暗号化されたビットストリーム

この発明の別の局面に従えば、ビットストリームは、２つの部分、すなわち暗号化できるまたはできないユーザの設計を表わすデータ部と、ビットストリームのロードを制御する（たとえば、ビットストリームの連続部分がロードされるＰＬＤの列のアドレスを与え、ローディング動作の信頼性を検査するための巡回冗長検査（ＣＲＣ）コードを与え、暗号ブロック連鎖（ＣＢＣ）、すなわち暗号化されたデータの発生頻度から解読されたデータを導出することができる「辞書攻撃」を防ぐ技術、の開始番号を与える）制御部とを含む。この発明の好ましい実施例では、データ部は暗号化されてもよいが制御部は暗号化されない。これは追加のセキュリティを与える、なぜならＰＬＤ製造業者はビットストリームの制御特徴を自由に記述する必要があり、この比較的よく知られた制御情報が暗号化された場合、攻撃者は、この情報を解読しこの情報を用いてビットストリーム全体を解読できるかもしれないからである。さらに、ビットストリームの制御部を暗号化されない状態にしておくことにより、ＰＬＤは情報を使用しやすくなる。

【００３６】

別の実施例では、構成データがロードされるアドレスの順序が、設計を解析するに当り攻撃者に役立つかもしれない場合に用いられ、構成データのアドレスも暗号化されるが、構成ビットストリームの他の制御情報は暗号化されないままである。

【００３７】

ビットストリームフォーマット

図４ａ～図４ｄは、図２ａ～図２ｄに示す先行技術の製品の構成ロジック１４のフォーマットおよびレジスタと比較した構成ロジック２９のビットストリームフォーマットおよびレジスタの違いを示す。図４ａに示すように、ビットストリームはさらに、ヘッダワードと、後続のデータワードとを含む。典型的な構成では、いくつかの構成データワードはレジスタにロードされ、その後に、暗号化された構成データが始まる。図４ａは、３つのヘッダワード、すなわちヘッダＡ、ヘッダＢおよびヘッダＣの各々の後に、３つの暗号化されていない制御データワード、すなわちデータＡ、データＢおよびデータＣが続く例を示す。（実際の構成では、３つより多い制御データワードが設けられる可能性が高い。）次に、ヘッダＤが、暗号化された構成データが後に続くことを明示し、その後には、暗号化された構成データの多数のワード、すなわちデータ１Ｄ、データ２Ｄ、データ３Ｄなどが続く。このデータが暗号化されていることを強調するために、これらのワードは図４ａにおいて斜線が付けられている。

【００３８】

図４ｂおよび図４ｃに示すように、４つの演算コードが追加される。演算なしの値００、解読なしの読出および書込の０１および１０に加えて、新しい値１１は、書込が解読つきであることを明示する（解読が用いられることや、それが演算コードにより特定されることを明示するために、どのようなコードまたはどのような方法が用いられるかは重要でない）。オプションの暗号化および解読が可能とされ示されることにより、設計者がこのオプションを利用できることだけが重要である。図４ｄの実施例では、２つの新しい構成論理レジスタが追加される。アドレス１１００および１１０１で示されるのは、暗号ブロック連鎖（ＣＢＣ）開始値を保持するレジスタと、初期暗号化鍵のアドレスとである。

【００３９】

オプションの暗号化

この発明の別の実施例に従えば、ＰＬＤはビットストリームの暗号化されたデータ部と暗号化されていないデータ部との両方を受付けることができる。ビットス

トリームの制御部は、ビットストリームのデータ部が暗号化されているかどうかを示す。ビットストリームのデータ部が暗号化されている場合、それはＰＬＤ内で解読器へと進路を変え、解読後にＰＬＤを構成するために用いられる。暗号化されていない場合、それは進路を変えず、ＰＬＤを構成するために直接使用される。

【００４０】

ビットストリームを暗号化しないことが好ましい場合がいくつかある。設計をデバックする際に使用されるある種のテスト動作は、構成情報をリードバックする必要がある。暗号化ステップが行なわれなかったならば（特に設計者が暗号化が問題と関係あるかどうかを判定しようとする場合）構成上の問題を診断することはより簡単である。また、数人の設計者が、ＰＬＤの幾つかの部分で実現されるコードを書いており、ＰＬＤの異なった部分が別々のときに構成される場合、ビットストリームのすべての部分を見える状態にし、ＰＬＤが部分的に再構成されることを可能にする必要があるかもしれない。

【００４１】

図５ａおよび図５ｂは、最初に暗号化されておらず次に暗号化される同じ設計を表わす例示のビットストリーム部を示し、この発明のある実施例における暗号化されていないビットストリームと暗号化されたビットストリームとの違いを示す。実際のビットストリームは、図の右側の０および１を含み、左側の文を含まない。左側の文は、右側のビットの意味を説明するために設けられている。これらのビットストリーム部は、図４ｂ～図４ｄに示すコマンドを用いる。図５ａの暗号化されていないバージョンと図５ｂの暗号化されたバージョンとの違いを強調するために、この違いは太字で示してある。

【００４２】

図５ａを見ると、ダミーワード（すべて１と解釈される一定のハイの信号である）および１および０の特定のパターンを有する周期ワード（ｓｙｎｃ　ｗｏｒｄ）の後の、次のワードは、１０の演算コードを有するタイプ００１であり、０００００００００００１００００のアドレスおよび０００００００００００１のワードカウントを有する。したがって、このワードは、コマンドレジスタＣＭＤを

アドレスし、１ワードがそこに書込まれることを特定する。図５ａにはビットストリームの左側に、このワードがＴｙｐｅ　１であることを示しかつｗｒｉｔｅ　１　ｗｏｒｄ　ｔｏ　ＣＭＤを示す注釈が付けられている。次のワード１１１は、コマンドレジスタＣＭＤに与えられるべきデータであり、ＣＲＣ（巡回冗長検査）レジスタをリセットする（好ましい実施例では、ＰＬＤは、ビットストリームがロードされているときにビットストリームからＣＲＣ値を計算するための、米国特許第５，５９８，４２４号においてＥｒｉｃｋｓｏｎによって記載されるような図示しない回路を含み、正しくないビットがロードされてしまいかねないビットストリーム電圧のグリッチから保護する）。次に、ヘッダワードは、フォーマットがやはりタイプ１であることを明示し、１ワードをフレーム長レジスタＦＬＲに書込むことを明示する。その後に続くデータワード１１００１は、フレーム長（２５ワード）を特定する。同様に、フレームアドレスレジスタＦＡＲに書込まれるべきワードを特定するヘッダを含む、いくつかの追加のヘッダおよびデータワードが続く。この場合には、後に続くデータワードは、データがアドレス０で始まることを示す。最後にこれらのレジスタがロードされてしまった後、コマンドはデータをフレームデータ入力レジスタＦＤＲＩに書込むようになり、多くのデータが書込まれるので、ワードカウントは０００００００００００として与えられ、タイプ２のヘッダは、１０５３０ワードがＦＤＲＩレジスタに書込まれることを特定する。これは、ＰＬＤが構成されるようにする実際の設計データである。したがって、ビットストリームにおける次の１０５３０ワードは設計データである。最後に、データが正しくロードされたことを保証するために、構成データを発生させたデバイスによって計算されたＣＲＣ値がロードされ、ＰＬＤによって計算されたＣＲＣ値と比較される。構成が完全なものであることを示し、かつＰＬＤを演算モードに移すために、さらなるコマンドおよびデータがロードされる。

【００４３】

図５ｂは図５ａと同様であるが、データおよび注釈が太字で示されている点のみが異なる。図５ｂにおいて、データは暗号化されており、追加のコマンドを用いて初期鍵アドレスを与え２ワード（６４ビット）をＣＢＣ（暗号ブロック連鎖）

レジスタに書込む。次に、タイプ１ヘッダは、演算コード１１を含み、フレームデータ入力レジスタＦＤＲＩに書込まれる前にデータが解読されることを示す。その後にはタイプ２ヘッダが続き、やはり演算コード１１を有しており、１０５３０ワードが解読されデータ入力レジスタＦＤＲＩに書込まれる旨の命令を与える。１０５３０個の暗号化されたデータワードが次に続く。次に、ＣＲＣワードが続き、（暗号化された）データが正しくロードされたことを確認する。最後に、追加のコマンドおよびデータが送信され、すべて正しいならばＰＬＤを演算モードにする。

【００４４】

解読プロセス

図６は、ある実施例においてオプションの解読が達成される態様を示す。図６は、構成ロジック２９と、解読器２４へ至るバス２７および２８との詳細を示す。バス２７は以下を含む：

・構成ロジック２９のレジスタアドレス１１０１（図４ｄ）から取得された３ビットの初期解読鍵アドレス「Ｉｎｉｔ＿ｋｅｙ＿ａｄｄｒ」

・６４ビットの修正された暗号ブロック連鎖値「ｍｏｄＣＢＣ」。この値は、構成ロジック２９のレジスタアドレス１１００（図４ｄ）から取得された６４ビットのＣＢＣ値の下位ビットをレジスタ０００１に特定される２２ビットのフレームアドレス値と置換することにより形成される。

【００４５】

・ビットストリームから取られ、解読されたデータをロードするための６４本の回線「Ｅｎｃｒｙｐｔｅｄ＿ｄａｔａ」

・解読器２４により生成された解読データを構成ロジック２９に返すための６４本の回線「Ｄｅｃｒｙｐｔｅｄ＿ｄａｔａ」

・データが「Ｅｎｃｒｙｐｔｅｄ＿ｄａｔａ＿ｌｉｎｅｓ」上にあることを解読器２４に示し、かつ解読器２４が解読を開始できることを示す信号のための回線「Ｅｎｃ＿ｄａｔａ＿ｒｄｙ」

・６４ビットのワード上の解読が完了し「Ｄｅｃｒｙｐｔｅｄ＿ｄａｔａ」上で利用可能であることを構成ロジック２９に示す信号のための回線「Ｄｅｃ＿ｄａ

ｔａ＿ｒｄｙ」

・解読器２４により使用され、構成ロジック２９に構成を中止させ、それにともない、たとえば鍵がある組の最初、中間または最後のものであるかを表わす鍵メモリ中のビットにより特定されるとおりに鍵が使用されていないとき、状態レジスタをセットするＢａｄ＿ｋｅｙ＿ｓｅｔ。図４ｄに示す実施例では、状態レジスタはアドレス０１１１にあり、Ｂａｄ＿ｋｅｙ＿ｓｅｔエラーは、論理１をビットの１つに記憶することにより示される。

【００４６】

バス２８は、以下からなる：

・最初にはバス２７から与えられる鍵アドレスであるが、新しい鍵が使用されるたびに更新される、鍵アドレスのための３本の回線

・解読鍵のための５６本の回線、および

・解読鍵が最初、中間、最後または使用される唯一の鍵であるかを示すための２本の回線。

【００４７】

設計再配置の防止

暗号化されたビットストリームにおける設計に対する攻撃の可能性の１つに、暗号化されたビットストリーム中のフレームアドレスレジスタ（開始アドレス）を変更して、それが解読されたとき、ＦＰＧＡが使用されているときに見えるＦＰＧＡの一部にロードされるようにすることがある。いくつかの設計では、ブロックＲＡＭの内容は見える。すべての設計において、入力／出力ポートの構成は見える状態になっており、そのため、構成ビットを判断することができる。したがって、設計の連続部分がＦＰＧＡの見える部分に移動された場合、ＦＰＧＡが適切に機能しなかったとしても、攻撃者は、再配置を繰り返すことで、暗号化されていないビットストリームの内容を知ることができる。

【００４８】

設計の再配置を防ぐために、ある実施例では、ＤＥＳ規格とともに用いられる暗号ブロック連鎖方法によって使用される初期値が修正される。図７ａおよび図７ｂは、この発明に従って修正された、三重ＤＥＳアルゴリズムの暗号化部と解読

部とをそれぞれ示す。標準の暗号ブロック連鎖方法は、開始番号（設計者が供給したものであっても、ランダムに生成されたものであってもよい）と暗号化されるべきデータの最初のワードとの排他的論理和を取ることにより暗号化プロセスを開始する。この発明に従えば、乱数の一部が、アドレス情報、この例では構成メモリ１２においてデータがロードされる最初のフレームの２２ビットのアドレスと置換される。開始ＣＢＣ値、すなわち６４ビットの数は、ｘと標識付けされた最下位ビットがｙと標識付けされたフレームアドレスと置換され、データがロードされるアドレスに依存する修正された６４ビット値を生成する。この修正されたＣＢＣ値は構成情報ワード１の最初のワードと排他的論理和が取られる。暗号化アルゴリズムを用いて最初の暗号化されたワード、すなわち暗号化ワード１を生成し、これはビットストリームに与えられる。図７ａは、外部暗号ブロック連鎖を有する三重暗号化アルゴリズムを示し、これは、第１の鍵を用いる暗号化ステップenc_1と、その後に続く第２の鍵を用いる解読ステップdec_2と、その後に続く第３の鍵を用いる暗号化ステップenc_3とを含む。この第１の暗号化されたワード、すなわち暗号化ワード１は、第２の暗号化されていないワード、すなわちワード２との排他的論理和が取られ、暗号化プロセスを繰返して暗号化されたワード２を生成する。すべての構成データが暗号化されてしまうまで排他的論理和連鎖は続く。

【００４９】

図７ｂに示すように、ＰＬＤは、解読されたワードを得るために逆のプロセスを行なわなければならない。上述の暗号化シーケンスについて、解読シーケンスは、鍵３を用いる解読ステップdec_1と、次に鍵２を用いる暗号化ステップenc_2と、次に鍵１を用いる解読ステップdec_3とである。重要なことは、解読ワード１を生成するための初期値の一部は、暗号化についても解読についても同じフレームアドレスを用いることである。ビットストリームではなく、ＰＬＤは、フレームアドレスレジスタに記憶されたフレームアドレスから修正されたＣＢＣ値を生成するが、これはまた、構成データがロードされる構成メモリ１２のフレームを特定するためにも使用される。したがって、攻撃者が、データがロードされるフレームアドレスを変更すると、修正されたＣＢＣ値がそれに伴って変

化し、構成データは正しく解読されない。

【００５０】

排他的論理和ステップは、暗号化される前は設計者のビットストリームにあった元々のデータを生成する。たとえば、元のワード１＝解読ワード１である。この解読された構成データは、バス２７（図３）で構成ロジック２９に送られる。

【００５１】

構成ロジック２９

構成ロジック２９は、オプションの暗号化をサポートする構造と、設計再配置および単一鍵攻撃を防止するための構造とを含む。図６に示すように、構成ロジック２９は、保持レジスタ２９２と、制御ロジック２９１と、構成レジスタ（ＦＤＲＩ、ＦＡＲ、ＣＲＣおよびｉｎｉｔＣＢＣが示される）と、解読器２４インターフェイスマルチプレクサ２９４および２９５と、６４ビットのアッセンブリレジスタ２９７と、レジスタ２９８および２９９（構成アクセスポート２２１とインターフェイスする）とを含む。６４ビットのシフトレジスタ２９９は、１ビット幅のデータのための単一のピンであっても、８ビット幅のデータのための８個のピンであってもよい構成アクセスポート２１からデータを受ける。レジスタ２９９がいっぱいになるまでこのデータは６４ビットのシフトレジスタ２９９にロードされる。そして、これらの６４ビットは、好ましくは、６４ビット転送レジスタ２９８に並列にシフトされる。そこから、マルチプレクサ２９６ｂが右側および左側の３２ビットワードを交互に選択し、マルチプレクサ２９６ａが、制御線Ｍによって制御されるように、保持レジスタ２９２へまたはアッセンブリレジスタ２９７の高位部分と低位部分とへ交互に、データを一度に３２ビットずつ移動する。ビットストリームのロードが始まると、回線Ｍおよび図示しないクロック信号により、マルチプレクサ２９６ａおよび２９６ｂは、６４ビット転送レジスタ２９８から保持レジスタ２９２にデータを移動する。そこから、これらのワードは制御ロジック２９１に与えられる。ワードがヘッダである場合、制御ロジック２９１はこのワードを解釈する。演算コードが、後続するデータが暗号化されずに書込まれるべきことを示す場合、制御ロジック２９１は、レジスタを選択するためにバスＧにアドレスを与え、回線Ｌに信号を与えてマルチプレクサ２９

４にバスＢからバスＤに接続させ、これに続くワードをバスＢに与える。次のクロック信号（クロック信号は図示しない）で、バスＤ上のデータはアドレスされたレジスタにロードされる。図４ｄに示すすべてのレジスタは、このようにロードされ得る。初期暗号ブロック連鎖値をロードするためのｉｎｉｔＣＢＣレジスタは、６４ビットのレジスタであり、図５ｂに示し上述したように２つの連続した３２ビットワードを受ける。

【００５２】

（１）ｉｎｉｔＣＢＣレジスタに記憶された元のＣＢＣ値と、（２）ＦＡＲレジスタに記憶された初期フレームアドレスとから形成された、修正されたＣＢＣ値は、解読器２４に利用可能となる。ある実施例では、ＦＡＲレジスタ中の初期フレームアドレスは、３２ビット以下を使用するが、ｉｎｉｔＣＢＣ値は６４ビットを用いる。図６の実施例では、修正されたＣＢＣ値を供給する６４ビットのバスは、フレームアドレスレジスタＦＡＲからの２２ビットと、ｉｎｉｔＣＢＣレジスタからの４２ビットとを含む。この発明によって提供されるセキュリティに重要なことには、この値は構成データがロードされるところに依存する。攻撃者が、ＦＡＲレジスタの内容を変えることにより、暗号化されたデータを異なった場所にロードしようとするならば、解読器２４に供給されるｍｏｄＣＢＣ値も変化してしまうだろう。

【００５３】

構成データの多数のワードを解読するための演算コードコマンドが制御ロジック２９１によって受けられると、解読プロセスが始まる。制御線Ｍにより、マルチプレクサ２９６ａは、転送レジスタ２９８からアッセンブリレジスタ２９７に至るバスＡにデータを与える。制御バスＨは、暗号化データレジスタ２９７の高位［３１：０］部および低位［３１：０］部とにバスＡを交互に接続して、解読されるべき６４ビットワードを形成する。制御ロジック２９１は次に、Ｅｎｃ＿ｄａｔａ＿ｒｄｙ信号をアサートし、これにより解読器２４はレジスタ２９７中のデータを解読する。

【００５４】

解読を行なうために、解読器２４は、鍵アドレスＫｅｙ　Ａｄｄｒをバス２８で

鍵メモリ２３（図３）に与える。これにより、鍵メモリ２３はそのアドレスの５６ビットの鍵を５６ビット鍵線に返す。また、鍵メモリ２３はそのアドレスの鍵データに記憶された２つの追加のビット「Ｏｒｄｅｒ」を返す。第１の解読鍵について、これらの２つのビットは、これが第１の鍵または唯一の鍵であることを示さなければならない。そうでない場合、解読器２４はＢａｄ＿ｋｅｙ＿ｓｅｔ信号をアサートし、これにより制御ロジック２９は構成動作を中止する。これらの２つのビットが、鍵が第１の鍵はまたは唯一の鍵であることを示す場合、解読器２４は、たとえば（Ｓｃｈｎｅｉｅｒにより同書に記載される）周知のＤＥＳアルゴリズムを用いて解読を行なう。鍵が唯一の鍵でなければ、解読器２４は鍵メモリ２３中の次のアドレスの鍵を獲得し、それが中間または最後の鍵かを２つのＯｒｄｅｒビットが示すかどうかを確認する。そうでなければ、Ｂａｄ＿ｋｅｙ＿ｓｅｔ信号がアサートされ構成が中止される。もしそうであれば、解読が行なわれる。それが中間の鍵であるならば、解読の別の一巡が行なわれる。それが最後の鍵であるならば、解読器２４は、解読されたワードと値ｍｏｄＣＢＣとの排他的論理和関数を形成する。解読器２４は次に、得られた値を６４ビットのＤｅｃｒｙｐｔｅｄ＿ｄａｔａバスに与え、Ｄｅｃ＿ｄａｔａ＿ｒｄｙ信号をアサートする。これにより、制御ロジック２９１は信号を制御線Ｋに与えて、マルチプレクサ２９５が６４ビットワードを２つの連続した３２ビットワードに分割するようにする。制御ロジック２９１は信号を回線Ｌに与えて、マルチプレクサ２９４が解読されたデータの３２ビットワードをバスＤに転送するようにする。制御ロジック２９１はまた、アドレス信号をバスＧに与えてフレームデータ入力レジスタＦＤＲＩをアドレスする。次のクロック信号が、解読されたデータをバスＥに移動させ、それはフレームレジスタにロードされ、フレームレジスタが一杯になると、ＦＡＲレジスタに示されるアドレスで構成メモリ１２に最終的にシフトされる。

【００５５】

ｍｏｄＣＢＣ値は、解読動作において１回のみ使用される。暗号化されたデータの後続の６４ビットワードが解読され、先に解読されたデータを用いて連鎖され排他的論理和演算を行なう。（ＦＡＲレジスタに記憶される値も、フレームアド

レスを選択するために１回のみ使用される。したがって、フレームが一杯になるたびにフレームアドレスはインクリメントされるだけである。）

動作の流れ

図８は、構成ロジック２９および解読器２４によって行なわれる動作の流れを示す。構成ロジック２９は、ビットストリームヘッダをロードし、対応するデータを図４ｂに示す構成論理レジスタに置くことにより、ビットストリーム長を判断することを含むステップ７０で始まる。ステップ７１で、起動シーケンスの後続の部分として、構成ロジック２９は、最初の構成メモリアドレスを読出す。ビットストリームフォーマットは、暗号化が用いられているかどうかを示す演算コードを含むことを想起されたい。ステップ７２は、演算コード値により分岐する。暗号化が用いられていない場合、この処理は図８の左側部分に示すとおりである。暗号化が用いられている場合、処理は図８の右側に示されるものである。暗号化がない場合、ステップ７３で、構成ロジック２９は、カウンタをビットストリームワードカウント（図４ｃ参照）に等しくセットする。ステップ７４で、構成データの３２ビット（１ワード）が構成メモリ１２のアドレスされたフレームに送られる。カウンタが終了したことをステップ７５が示す場合、ステップ７６でカウンタがデクリメントされ、構成データの次の１ワードが構成メモリ１２に送られる。カウンタが処理を終了した場合、構成ロジック２９は、最終巡回冗長値を読出してビットストリームの最後の値と比較しビットストリームのロードにエラーがあったかどうかを判定することを含むクリーンナップ動作を行なう。

【００５６】

ビットストリームが暗号化されていることをステップ７２が示す場合、カウンタにはワードカウントがロードされ、ステップ８１で、この処理は、初期鍵アドレスを鍵アドレスレジスタ２９３（図６）から解読器２４にロードする。

【００５７】

ステップ８２で、暗号化された構成データの２ワード（６４ビット）が解読器２４にロードされる。ステップ８３で、アドレスされた鍵が解読器２４にロードされる。ある実施例では、６４ビットの数が解読器２４にロードされる。この６４ビットの数は、５６ビットと、それが最初の鍵か、中間の鍵か最後の鍵かまたは

唯一の鍵であるかを示す２ビットと、未使用であってもパリティのために使用されても別の目的のために使用されてもよい、いくつかの他のビットとを含む。別の実施例では、６４ビットの鍵データは、それが最後の鍵であるか否かを示す単一ビットを含む。さらに別の実施例では、６４ビット鍵データが次の鍵のアドレスを含むので、鍵はシーケンシャルな順序で使用されなくてもよい。別の実施例では、追加ビットが存在せず、鍵データは６４ビット未満を用いる。さらに別の実施例では、鍵ではなくビットストリームが、いくつの鍵が使用されるべきかを示すが、これは、安全性が低いと考えられる。というのも攻撃者が、いくつの鍵が用いられるかがわかり単一鍵攻撃を行なって、一度に１つの鍵を解読することができるからである。これに対し、いくつの鍵が使用されるべきかを示す鍵を用いれば、この情報は攻撃者に漏れない。

【００５８】

ステップ８４で、解読器２４は、たとえばＤＥＳアルゴリズムを用いて５６ビットの鍵で６４ビットのデータを解読する。ＤＥＳアルゴリズムは、Ｂｒｕｃｅ Ｓｃｈｎｅｉｅｒによる上述の本の２６５から２７８頁に記載されている。たとえば、高度暗号化規格ＡＥＳなどの他の暗号化アルゴリズムを用いてもよい。他のアルゴリズムはより多くの鍵ビットを必要とするかもしれない。たとえば、ＡＥＳは１２８から２５６ビットの鍵を必要とする。

【００５９】

ステップ８５は、より多くの鍵が使用されるかどうかを判断する。鍵が最初、中間、最後または唯一の鍵であるかを示す２ビットを調べて、これが最後の鍵であるかどうかを判定し、そうでなければ、鍵アドレスがインクリメントされ、解読器２４はメモリ２３中の次の鍵をアドレスする。

【００６０】

最後の鍵が使用された後、ステップ８７で、レジスタＦＡＲおよびｉｎｉｔＣＢＣを組合わせることから６４ビット値として図６に示す修正されたＣＢＣ値と、ステップ８７で得られた解読された値との排他的論理和がとられる。ある実施例では、ＣＢＣレジスタにロードされる６４ビットの乱数のうち２２ビットが、ビットストリームの始まりのフレームアドレスと置換される。この暗号化処理の目

標は、６４ビットの暗号化された値のあらゆる桁を、鍵に加えてすべての前のビットの関数とすることである。ＣＢＣ値を最初のアドレスと組合せることの目標は、ビットストリームが意図された開始アドレスと異なるアドレスにロードされる場合、解読された値が変わるようにすることである。ステップ８７は、この目標の両方を達成する。次に、新しいＣＢＣ値が記憶される。図６に示すＦＡＲレジスタおよびＣＢＣレジスタに記憶されても、または解読器２４にある別のレジスタに記憶されてもよい。

【００６１】

ステップ８８で、この解読された構成データはバス２７（図３）で構成ロジック２９に送られる。構成ロジック２９は、更新された巡回冗長検査値を計算し、ロード処理の終りにＣＲＣレジスタに記憶された巡回冗長値と比較する。構成ロジック２９が暗号化を用いるように設定されているならば、構成ロジック２９のマルチプレクサは、この解読された構成データを構成メモリ１２のアドレスされた列に転送する。

【００６２】

ステップ８９で、カウンタが検査され、もし終了していなければ、ステップ９６でカウンタはデクリメントされ、処理はステップ８２に戻り、ここで次の６４ビット（２ワード）がビットストリームからロードされる。

【００６３】

最後に、カウンタが終了したことをステップ８９が示す場合、ステップ９０で、ビットストリーム中のＣＲＣ（巡回冗長検査）値が、ビットストリームがロードされるときに計算されたＣＲＣ値と比較される。これらの値が一致するならば、構成は完成しておりＦＰＧＡは動作に入る。これらの値が一致していなければ、ロードの際のエラーが発生しており、構成処理全体が中止される。

【００６４】

鍵の順序の評価－単一鍵攻撃の防止

図９は、鍵の順序を評価するために解読器２４によって実現される状態マシンを示す。状態マシンは、Ｅｎｃ＿ｄａｔａ＿ｒｅａｏｌｙ信号が活性化されるまで状態Ｓ１にある。この信号は、解読が開始可能で決定状態Ｑ１に移ることを示

す。決定状態Ｑ１では、解読器２４は、バス２７上のアドレスＩｎｉｔ＿Ｋｅｙ＿ａｄｄｒにより特定されるアドレスをバス２８に与え、鍵および鍵の順序をリードバックし、２ビットの鍵順序データから、鍵が最初または唯一の鍵であるかを判定する。そうでなければ、解読器２４は、Ｂａｄ＿ｋｅｙ＿ｓｅｔ信号を制御ロジック２９１に送り、制御ロジック２９に構成を中止させる。アドレスが最初または唯一のものであれば、解読器２４は状態Ｓ３に入り、データを解読する。そして、状態マシンは決定状態Ｑ２に入り、鍵が最後または唯一のものかを判断する。もしそうであれば、解読が完了し状態Ｓ４で解読器２４は解読されたデータを構成ロジック２９に返す。そうでなければ、状態Ｓ５で、解読器２４は鍵アドレスをインクリメントし、新しい鍵を獲得する。状態マシンは、次の鍵が中間または最後の鍵であるかを判断するための質問Ｑ３を尋ねる。そうでなければ、状態Ｓ２は構成を中止させる。鍵が中間または最後のものであれば、状態マシンは状態Ｓ３に戻りデータを再び解読する。別の実施例では、状態Ｓ４において、解読器２４は、ＣＢＣ値と解読されたデータとの排他的論理和をとるステップも行なう。

【００６５】

鍵の中に鍵の順序を記憶することの利益は、攻撃者が単一鍵攻撃を実現できないことである、というのも、攻撃者は、解読を行なう際、解読器２４が（設計者によって意図される）鍵メモリ２３によって特定される鍵のすべてを用いることを防ぐことができないからである。鍵の順序がＰＬＤ内の鍵データ内に記憶されるので、第２の質問Ｑ２および第３の質問Ｑ３を尋ねて単一鍵攻撃を使用する攻撃者から保護する必要はない。しかしながら、有利には、鍵をロードする設計者または基板検査師は、３つの質問のすべてを尋ねて各鍵がロードされる際に確実に正しく標識付けされるようにする。

【００６６】

ある実施例では、解読器２４は、解読－暗号化－解読シーケンスを備える三重ＤＥＳ規格を用い、別の鍵が使用されるたびにアルゴリズムを（ぼんの少しだけ）交互にする。そのような組合せは、ＡＮＳＩ　Ｘ９．５２　１９９８三重ＤＥＳ規格に従っている。別の実施例では、毎回解読が使用される。

【００６７】

鍵メモリ２３

図１０ａに示す回路は、３つの構成要素、すなわち電池供給スイッチ２２と、制御ロジック２３ａと、鍵レジスタ２３ｂとを含む。制御論理回路２３ａおよび鍵レジスタ２３ｂは図３の鍵メモリ２３を含む。図１０ａの実施例では、鍵レジスタ２３ｂは６個の６４ビットワードを含む。もちろん、他の鍵メモリの寸法がこれに代えて使用されてもよい。他の実施例では、鍵メモリ２３に記憶される鍵は６つよりはるかに多くてもよく、使用される鍵のアドレスを与えるためには３ビットより多くが必要とされる。鍵レジスタ２３ｂの電源は、ラインＶＳＷＩＴＣＨにより電池供給スイッチ２２から来る。鍵メモリ供給電圧ＶＣＣＩが不十分か存在していない場合、電池供給スイッチ２２が電池バックアップ電圧ＶＢＡＴＴをＶＳＷＩＴＣＨラインに与え、これによりＶＳＷＩＴＣＨは正の電圧を担持する。

【００６８】

この実施例では、各鍵レジスタは６４個のメモリセルを有する。各セルはライトイネーブル信号ＷＥを受け、これは高レベルのとき、データがセルに書込まれ、低レベルのときセルのデータが保持される。あるレジスタにおけるセルは共通のライトイネーブル信号ＷＥを有する。ＰＬＤ電源電圧（ＶＣＣＩと異なる）がなくそのためＷＥ信号が活性に駆動されない場合、Ｔ１などの弱いプルダウントランジスタがＷＥ信号をプルダウンして、いずれの鍵メモリレジスタもアドレス不可能とし、いずれのメモリセルも妨害されないようにする。

【００６９】

ある実施例では、ＰＬＤのＪＴＡＧポートは、解読鍵をＰＬＤにロードするために用いられる。メモリセル電源電圧は、通常動作時にはＶＣＣＩのデバイス電圧レベルにあり、ある実施例では、このレベルは３．０と３．６ボルトとの間である。ＪＴＡＧポートに与えられる信号は幾つかの異なった電圧であってもよい。また、いくつかの異なった内部電圧があってもよい。したがって電圧変換が必要である。この電圧変換はメモリセルにおいて行なわれる。メモリセルの詳細は図１０ｂに示される。インバータＩ１およびＩ２を含むラッチには、ＶＳＷＩＴＣ

Ｈによって電源が供給され、このためデバイス電源電圧ＶＣＣＩがあってもなくても電源が供給される。ＷＥ信号および反転されたデータ信号ｄａｔａ＿ｂはどちらも１５ボルトのレベルで作用する。これらの信号は、ＮＭＯＳトランジスタＴ４、Ｔ５およびＴ６を駆動し、インバータＩ３を介して（１．５ボルトの電源電圧も用いる）トランジスタＴ７を駆動する。図１０ｂは、ＷＥが低レベルのときトランジスタＴ４およびＴ５がオフであることを示し、インバータＩ１およびＩ２を含むラッチの内容が保持される。ＷＥが高レベルのとき、インバータＩ１およびＩ２の一方が低レベルになり、新しいデータをラッチにロードする。

【００７０】

制御論理回路２３ａはＪＴＡＧバス２５（図３にも示す）から信号を受ける。ＪＴＡＧバス２５は、書込、読出、安全モードを設定するための制御信号ならびにデータバスおよびアドレスバスを含む。このインターフェイスは、ＩＥＥＥ１５３２ＪＴＡＧ規格に準拠している。鍵メモリ２３がＪＴＡＧバス２５を介してアクセス可能となる前に、セキュリティ状態（バス２６）が非安全モードにされるが、これは、ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令（図１０ａを参照）を用いて鍵データバスのビット０に論理１を与えることで行なうことができる。鍵メモリ２３は、ＩＥＥＥ１５３２規格のＩＳＣ＿ＰＲＯＧＲＡＭ命令およびＩＳＣ＿ＲＥＡＤ命令を用いてＪＴＡＧバス２５に書込まれＪＴＡＧバス２５から（検証のために）読出される。制御ロジック２３ａは、ＪＴＡＧバス２５から３ビットのアドレス信号ＡＤＤＲをデコードするデコーダを含み、ＩＳＣ＿ＰＲＯＧＲＡＭ命令がＪＴＡＧバス２５に現われる場合、書込ストローブ線ｗｓ＿ｂ［５：０］のうちアドレスされたものに低レベルになるパルスを生成し、またはＩＳＣ＿ＲＥＡＤ命令がＪＴＡＧバス２５に現われる場合、読出選択線ｒｓｅｌ［５：０］のうちアドレスされたものに高レベルの信号を生成する。６個の６４ビットワードのうちの１つは、高レベルの信号を６本の読出選択線ｒｓｅｌ［５：０］の１本に与え、読出マルチプレクサ２３ｄが選択されたワードを６４本の出力線ｑ［６３：０］に与えることにより、読出すことができる。書込選択線または読出選択線のうち１つのみが一度に選択される。どの読出選択信号もアサートされない場合、高レベルのｐａｒｋ＿ｌｏｗ信号により、６４個のトラン

ジスタ２３ｅが６４本の線ｑ［６３：０］をプルダウンし、これらの回線が浮遊するのを防ぐ。

【００７１】

鍵メモリ２３が非安全モードで作動している場合、６４ビットワードを鍵レジスタ２３ｂからＪＴＡＧバス２５に読出すことができ、ＦＰＧＡの外部で値を調べることができる。レジスタ２３ｂ中の選択された６４ビットワードのうち５６ビットをＤＥＳ解読のための５６ビットの鍵として用いることにより、ＦＰＧＡをこの非安全モードでテストすることができる。ある実施例では、鍵メモリ２３が非安全モードにあるとき、設計がロードする前に暗号化されていたとしてもユーザの設計のリードバックは可能である。これにより、設計者は暗号化された設計でもテストおよびデバッグすることができる。鍵セキュリティ状態の通信はバス２６を介している（図３も参照）。

【００７２】

値が鍵レジスタ２３ｂに書込まれバス２５からの読出動作で確認された後、制御ロジック２３ａは、ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令を用い、ＩＥＥＥ１５３２規格の一部である６４ビットの鍵データバスのビット０に論理０を与えることにより安全モードに置かれる。安全モードでは、いかなる鍵へのアクセスも許されない。

【００７３】

図１１に示すように、ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令を用いて鍵を読出すことにより確実に攻撃者が非安全モードに戻ることができないように、セキュリティが除去された場合（ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ信号が非安全論理レベルに移る場合）、制御ロジック２３ａの状態マシンは、一度に１ワードずつ、すべての６個のワードに０を書込むことによりすべての鍵を消去する。これは、ステップ１１０で、０をｗｄａｔａ［６３：０］バスに与え、ステップ１１１でｗｓ＿ｂ［０］信号を（論理０値で）アサートし、次にステップ１１２～１１７で一度に１つずつｗｓ＿ｂ［０：０］信号からｗｓ＿ｂ［５：０］信号を連続的にストローブし、その後ステップ１１８でセキュリティ状態を変更して非安全モードに入り、最後にステップ１１９でｗｄａｔａ［６

３：０］論理０値を解放することによりなされる。したがって、電池でバックアップされたメモリ２３を非安全モードにしようとすると、鍵レジスタ２３ｂにある値のすべてが消去されてしまう。

【００７４】

鍵メモリ２３が安全モードにあるかどうかを伝達するために、制御ロジック２３ａは、鍵メモリ２３が安全モードで動作していることを示すために安全モード信号をバス２６（単一の線であってもよい）で構成ロジック２９に送る。この信号が非安全モードに切換わるならば、構成ロジック２９は構成メモリ１２から設計をクリアする。なお、鍵が鍵レジスタ２３ｂに記憶されておりかつ鍵メモリ２３が安全モードにあるとしても、暗号化されていないビットストリームは構成ロジック２９により構成メモリ１２にロードされ得る。

【００７５】

鍵のロード、複数の暗号化鍵

ユーザが設計の詳細を知ることのできない安全モードにＰＬＤが入る前に、解読鍵はＰＬＤにロードされなければならない。図３に示す実施例では、鍵はＪＴＡＧポートを介してロードされる。

【００７６】

この発明の特徴として、暗号化鍵はこのＪＴＡＧポート２０を介してロードされる。ＪＴＡＧプログラマは基板のテスト中に暗号化鍵をロードすることが求められる。鍵を記憶するためのＲＡＭが非安全モードにあるとき、ユーザはそれへの完全なアクセスを有しており、設計が暗号化されていたとしても鍵および設計の両方を読出すことができる。鍵をテストしながら鍵を使用できるのでこれは設計者にとって有用である。そして、設計者が動作に満足すると、別の命令をＪＴＡＧポートを介して送り、鍵メモリを安全モードにすることができる。鍵メモリが安全モードになってしまうと、鍵は読出すことができない。さらに、鍵メモリを安全モードから非安全モードに移すことで、メモリ初期化処理を起動する回路を活性化することにより鍵が消去される。（以下に記載する図１５は、この機能を行なうための状態マシンを示す。）

この発明のある局面に従えば、設計を暗号化するのに２つ以上の鍵が用いられて

もよい。たとえば、３つの鍵が用いられる場合、まず第１の鍵を用いてビットストリームを暗号化し、第２の鍵を用いて得られた暗号化されたビットストリームを再び暗号化し、最後に第３の鍵を用いて得られた二重に暗号化されたビットストリームを再び暗号化する。この三重に暗号化されたビットストリームは、たとえば、ＰＬＤを保持するプリント回路基板上のＰＲＯＭまたはフラッシュメモリに記憶される。

【００７７】

解読については、これらの鍵を連続して（逆の順序で）用いて、暗号化されたビットストリームを繰返し解読する。さらに、特定の設計を解読するのに使用されるよりも多くの鍵がＰＬＤに記憶されている場合、暗号化されたビットストリームは、暗号化されていない部分に、いくつの鍵が使用されるべきかの指示と、第１の鍵のアドレスとを含み得る。そのような実施例では、攻撃者は一度に１つの鍵に対処するだけでよいので、攻撃者は容易にビットストリームを解読できてしまうかもしれない。代替的に、鍵自体が、それらが最初の鍵か、中間の鍵か、最後の鍵か、または唯一の鍵であるかを示してもよい。したがって、同じＰＬＤを、異なった機能（異なった設計で構成される）を行なうように何回かに分けてプログラムすることができ、異なった鍵の値についての情報を、設計者の１人にのみまたは幾人かに利用可能とすることができる。したがって、どちらの設計も同じＰＬＤに実現される（何回かに分けて）としても第１の設計者が第２の設計について知らないことがあり得る。

【００７８】

図３に関し、構成ロジック２９は、図１の構成ロジック１４を超える追加のロジックを含む。図１の構造におけるように、構成アクセスポート２１上のビットストリームは、ワードとして、ある実施例では３２ビットのワードとして扱われる。通常、ビットストリームの開始におけるまたはその付近におけるワードのうちのいくつかが、ヘッダ情報、たとえばビットストリームの長さ、構成データの開始アドレスを含む。この発明のビットストリームにとって新しいことは、ビットストリームが暗号化されているかどうかについての指示と、ビットストリーム中の構成データを解読するための鍵のアドレスとである。

できるが、攻撃者は鍵に記憶された値を知ることができない。したがって、暗号化されていない設計を読出したり複製したりすることはできない。このセキュリティを達成するために、いくつかのステップがとられる。

【００８７】

安全モードの保存（改ざんの証明）

ある実施例では、ＰＬＤの構成ロジック２９には２つのセキュリティフラグがある。一方は、解読鍵が安全にされたかどうかを示し、他方は、設計が解読された設計であり保護されなければならないかどうかを示す。ＪＴＡＧロジック１３（図３）がＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令で安全モードを選択する場合、制御ロジック２３ａ（図１０ａ）のｓｅｃｕｒｅ＿ｋｅｙフラグがセットされる。ＰＬＤにロードされるビットストリームが、そのビットストリームの設計データが暗号化されているという指示を有するならば、構成ロジック２９（図示せず）のｓｅｃｕｒｅ＿ｄｅｓｉｇｎフラグがセットされる。いずれかのフラグが後に設定解除されるならば、構成メモリの全体がクリアされ、解読された設計が除去される。ｓｅｃｕｒｅ＿ｋｅｙフラグが（ＩＳＣ＿ＰＲＯＧＲＡＭ＿ＳＥＣＵＲＩＴＹ命令により）リセットされた場合、鍵も消去される。

【００８８】

図１５は、設計クリア機能を行なうための状態マシンを示す。ｓｅｃｕｒｅ＿ｄｅｓｉｇｎフラグがセットされると、状態マシンは状態Ｓ１にはいる。この状態は、ｓｅｃｕｒｅ＿ｄｅｓｉｇｎフラグの安全モードから非安全モードへの変化をモニタする。安全設計モードが持続する限り、状態マシンは状態Ｓ１に留まる。変化が起きると、状態マシンは状態Ｓ２に入り、データを構成メモリ１２にシフトするためのデータシフトレジスタがリセットされ、構成メモリビットのためのすべてのデータ線に０が与えられる。次に、状態マシンは、アドレスされたフレームのワード線がアサートされる状態Ｓ３に移る。この結果、データシフトレジスタ線上の０が、アドレスされたフレームのメモリビットに書込まれる。質問Ｑ１がアドレスされるべきフレームがさらにあることを示す場合、状態マシンは、フレームアドレスが進められる状態Ｓ４に移り、状態マシンは状態Ｓ３に戻る。質問Ｑ１がアドレスされるべきフレームがこれ以上ないことを示すとき、処

理は終わり構成メモリがクリアされる。

【００８９】

鍵が攻撃者によってアクセスされるのを保護する必要もある。設計を含むシステムが末端の消費者に利用可能となる前に鍵のロードが行なわれる。設計者が設計を開発する処理をしているとき、彼らは、非安全モードでＰＬＤを動作させてデバッグしたいことがある。このデバッグ操作を可能とし、かつ鍵のセキュリティを保護するために、鍵ロード処理は、すべての鍵レジスタをクリアすることにより非安全モードで始まる。鍵がロードされている間および検証のため鍵がリードバックされている間、安全鍵フラグは非安全モードに維持されなければならない。構成ビットストリームがロードされ解読されている間も、安全鍵フラグは非安全モードに維持されなければならない。しかし、安全鍵フラグが一旦セットされると、安全鍵フラグを非安全モードに戻すことにより、すべての鍵をクリアし、また図１５の状態マシンの動作を起動する。そのため、鍵がクリアされるのみならず構成もクリアされる。

【００９０】

リードバック攻撃およびリードバック不能化

いくつかのＦＰＧＡは、ビットストリームがＦＰＧＡからリードバックされることを可能としており、そのためユーザは設計をデバッグしたりＦＰＧＡのフリップフロップから状態マシン情報を獲得し得る。設計がリードバック動作のために再暗号化されないとすれば、ビットストリームをリードバックする動作が、暗号化されていないビットストリームを見えるように暴露してしまうだろう。

【００９１】

設計のなお一層のセキュリティは、暗号化された設計がＦＰＧＡにロードされるときリードバックを不能化することにより提供される。ある実施例では、解読鍵も安全にされた場合にのみリードバックが不能化される。

【００９２】

図１６は、構成メモリをロードしリードバックするための構造のブロック図を示す。ある実施例では、構成ロジック２９は、以下の２つの条件が存在するときリードバックを防ぐ：（１）データバス２６（図３および図１０を参照）上のセキ

ュリティ状態線が鍵が安全モードにあることを示し、かつ（２）構成ロジック２９が、ビットストリームが暗号化されていることを示す構成ビットストリーム中の演算コードに応答したときである。両方のキーが安全にされていないかまたはビットストリームが暗号化されていない場合に、リードバックを可能化することができる。他の実施例では、異なった条件が、リードバックを可能化するかどうかを制御する。

【００９３】

構成ロジック２９が、リードバックが行なわれるべきことを示すヘッダをビットストリーム中に受けるとき、それは、フレームアドレスレジスタに記憶されたフレームアドレスを線１０７に送り、このフレームアドレスは、バス１０９のアドレスされた線を選択するためにアドレスデコーダ１１０によってデコードされる。次に、線１０８上のワード線可能化信号がアサートされ、これがバス１０９の選択されたワード線をアサートし、選択されたワード線によりアドレスされるメモリセルがその値をｎ本のデータ線１０２に与える（ｎはフレーム長であり構成ロジック２９に記憶される）。そして、構成ロジック２９はライン１０４上のロード信号をアサートして、データのフレームを（並列に）データシフトレジスタ１０１にロードする。次に、構成ロジック２９は、線１０５上のシフト信号をアサートして、これによりデータシフトレジスタ１０１は、バス１０３上の３２ビットのワードのデータのフレームをフレームデータ出力レジスタ（図４ｄ）にシフトアウトし、そこから構成アクセスポート２１（図３）上の出力ビットストリームにシフトアウトする。

【００９４】

解読がビットストリームに示される場合、構成ロジック２９は内部フラグをセットしてこれを示す。これらのフラグがセットされ、かつ鍵メモリ２３がバス２６上のセキュリティ状態信号により示されるように安全モードにある場合、構成ロジック２９は、回線１０８上のワード線可能化信号を不活性に保ち、かつ回線１０４および１０５上のロード信号およびシフト信号を不活性に保ってリードバックを防ぐことにより、ビットストリーム中のリードバックコマンドに応答する。しかしながら、鍵メモリ２７が安全モードになければ、設計が暗号化されていた

としても、リードバックは許可され、テストおよびデバッグが可能となる。

【００９５】

部分的再構成攻撃および防止

いくつかのＦＰＧＡは、ＦＰＧＡの部分的再構成を可能にしたり、別々の開始アドレスおよび別々の書込命令を用いて設計の異なった部分がＦＰＧＡの異なった部分にロードされることを可能にする。攻撃者は、設計を部分的に再構成してブロックＲＡＭまたはフリップフロップの内容を出力ポートに直接読出したり、またはセクションを既存の設計に追加して、設計を知るために使用できる情報を読出すことにより、設計を知ろうと企てるかもしれない。たとえば、攻撃者は、ＰＬＤを暗号化されていない設計で部分的に再構成するかもしれないが、その唯一の目的は暗号化された設計についての情報を抽出することである。そのようなトロイの木馬設計は、別のビットストリームでＰＬＤにロードされるか、既存の暗号化されたビットストリームに付加されるかもしれない。たとえば、攻撃者が、ＦＰＧＡのブロックＲＡＭにロードされた状態マシン設計を知ることに興味を持った場合、トロイの木馬設計は、ブロックＲＡＭのアドレスを循環しブロックＲＡＭデータの内容をパッケージピンに送るロジックを含むかもしれない。

【００９６】

攻撃者がそのような変更を加えることを防ぐために、当初の設計が暗号化される場合、構成ロジック２９は、解読つきの構成が一旦起動されると部分的再構成を禁じる。構成ロジック２９は、解読演算コードを有するヘッダが処理されてしまうと後続の書込命令を禁じる。また、構成ロジック２９は、暗号化なしの構成が行なわれると解読付きの構成を禁じる。構成ロジック２９は、解読命令が受信された後、構成メモリへの書込を行なうヘッダを無視し、設計の暗号化されていない部分がロードされてしまった場合、解読コマンドを有するヘッダを無視することにより、これらの制限を達成する。したがって、いずれかの演算コードが解読付きの書込が用いられていることを示すならば、ＰＬＤは単一の書込命令のみを受付ける。

【００９７】

追加の実施例

図面の上述の説明はいくつかの実施例についての詳細を与えた。しかしながら、多くの追加の実施例も可能である。たとえば、上述した暗号ブロック連鎖アルゴリズムの代わりに、ブロックサイズよりも小さいユニットで、たとえば一度に１個の８ビットバイトでデータを暗号化することのできる暗号フィードバックモードと呼ばれる暗号化方法を用いることができる。この暗号フィードバックモードは、Ｓｃｈｎｅｉｅｒにより同書の２００から２０３頁に記載される。

【００９８】

さらに別の実施例では、暗号化が用いられる場合、アドレス０で始まってすべてのビットストリームがロードされなければならない。この実施例のある実現化例は、暗号化を特定する演算コードが受信されたとき、開始フレームアドレスレジスタＦＡＲ（図６）にロードされたアドレスをアドレス０と置換する。

【００９９】

さらに別の実施例では、開始アドレスおよび設計データの両方を暗号化する。この実施例では、暗号化されていない設計データで可能なように、異なったフレームアドレスで始まる暗号化された設計データのいくつかのセグメントをロードすることが可能である。

【０１００】

別の実施例では、鍵メモリ２３などの鍵メモリに記憶される鍵データは、後に続く鍵の数を明示する。この実施例の変形例では、鍵データは、その鍵に先行する鍵の数も明示する。もし攻撃者が、設計者が意図した最初の鍵アドレスと異なる鍵アドレスを与えるならば、構成が中止されてもよい。さらに、鍵内に明示される数の鍵が使用されてしまうまで、暗号化が行われる。

【０１０１】

別の実施例では、鍵メモリが非安全モードにあるとき鍵がリードバックされることを可能にする代わりに、鍵は、パリティビットまたはＣＲＣチェックビットを含み、これらのビットのみをリードバックして鍵が正しくロードされたかことを確認することができる。この実施例は、ある設計者に知られている鍵を別の設計者に秘密にすることができ、ＰＬＤが異なった設計をロードするために複数回に分けて使用されるときに有用である。

【０１０２】

上述のＣＲＣチェックサム計算に関し、ＣＲＣチェックサムを、設計が暗号化される前または後のいずれかに計算する実施例が提供され得る。もちろん、設計データが暗号化される前に、ビットストリームに付加されたチェックサムが計算されるならば、対応のチェックサムは、解読された後に設計データに対してＰＬＤ内で計算されなければならない。同様に、設計データが暗号化された後に、ビットストリームに付加されたチェックサムが計算されるならば、ＰＬＤは、設計データが解読される前に受信ビットストリームに対して対応のチェックサムを計算しなければならない。

【０１０３】

解読鍵をロードする処理に関し、図８に示す処理を用いる場合、デバイスプログラマを用いて解読鍵をロードする必要はない。鍵は、基板検査手続の一部としてロードされればよい。

【０１０４】

２つ以上のＰＬＤをプログラムするために上述の構造および方法を用いることも可能である。いくつかのデバイスをデイジーチェーン接続に構成しビットストリームをデバイスに直列に通過させるか、またはデバイスを直列にアドレスすることにより、２つ以上のＰＬＤまたはＦＰＧＡをプログラムするために単一のビットストリームを用いることは周知である。デバイスのうち１つ以上が暗号化された設計データを受信するものである場合いくつかのＰＬＤをそのような構成に配置することが可能である。

【０１０５】

さらに別の実施例として、暗号化された設計データを有するビットストリームに対して単一のアドレスのみが特定されてもよい実施例が記載されたが、別の実施例では、好ましくは暗号化されたいくつかのアドレスが、設計の別々の部分をロードするために特定されてもよい。さらに、これらの別々の部分は同じ暗号化鍵を用いてもよく、またはこれらの別々の部分は、異なった暗号化鍵および鍵の異なった組を用いてもよい。

【０１０６】

以上の説明から明らかとなる変形はこの発明の範囲に含まれるものとする。

【図面の簡単な説明】

【図１】先行技術のＦＰＧＡにおける機能的関係の図である。

【図２ａ】先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図２ｂ】先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図２ｃ】先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図２ｄ】先行技術のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図３】この発明のある実施例に従うＦＰＧＡにおける機能的関係の図である。

【図４ａ】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図４ｂ】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図４ｃ】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図４ｄ】この発明のビットストリームに含まれ得るビットストリームフォーマットおよびコマンドの図である。

【図５ａ】例示の暗号化されていないビットストリームおよび暗号化されたビットストリームの図である。

【図５ｂ】例示の暗号化されていないビットストリームおよび暗号化されたビットストリームの図である。

【図６】構成ロジック２９ならびに解読器２４に至るバス２７およびバス２８における線の図である。

【図７ａ】この発明のある実施例に用いられる三重暗号化での外部暗号ブロック連鎖のための修正された開始値の図である。

【図７ｂ】図７ａで使用される対応の開始値および解読処理の図である。

【図８】ビットストリームを処理するための操作の図である。

【図９】鍵の順序を評価するために解読器２４によって実現される状態マシンの図である。

【図１０ａ】図３の鍵メモリ２３の構造の図である。

【図１０ｂ】図１０ａのメモリセルの構造の図である。

【図１１】非安全とされたときに鍵を消去するために図１０ａの制御ロジック２３ａによって行なわれるステップの図である。

【図１２】図１０ａの電池供給スイッチを詳細に示す図である。

【図１３】図１２の電池供給スイッチのレベルシフト回路および電圧検出回路の図である。

【図１４】図１２の電池供給スイッチのレベルシフト回路および電圧検出回路の図である。

【図１５】安全モードから抜けたとき設計を消去するための状態マシンの図である。

【図１６】暗号化が存在するとき不能化される線を含む、構成メモリをロードし構成をリードバックするための要素のブロック図である。

【図１】

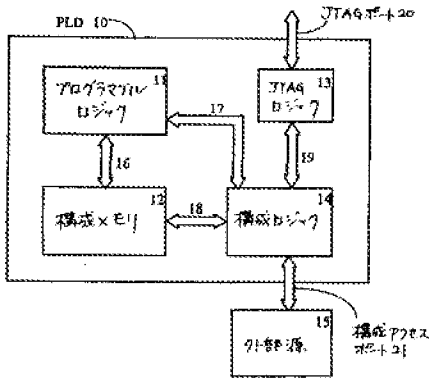


Fig. 1
PRIOR ART

【図２ａ】

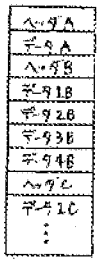


Fig. 2a
PRIOR ART

【図２ｂ】

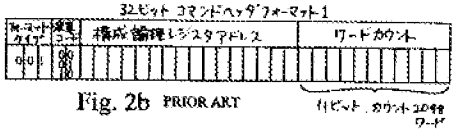


Fig. 2b PRIOR ART

【図２ｃ】

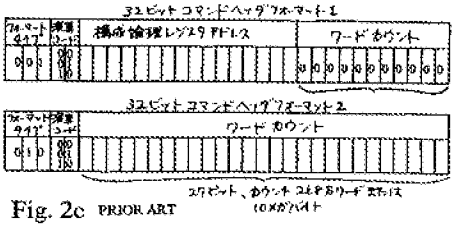


Fig. 2c PRIOR ART

【図２ｄ】

ビットストリームヘッダ構成論理レジスタアドレス	構成論理レジスタランタ内容
0000	コマンド完了 状態 (CRC)
0001	フレームアドレス
0010	フレームデータ入力
0011	フレームデータ出力
0100	コマンド
0110	予備
0111	ステータス
1000	デバイスID
1001	構成オプション
1010	予備
1011	フレームB
1100	予備
1101	予備
1110	予備
1111	予備

Fig. 2d
PRIOR ART

【図３】

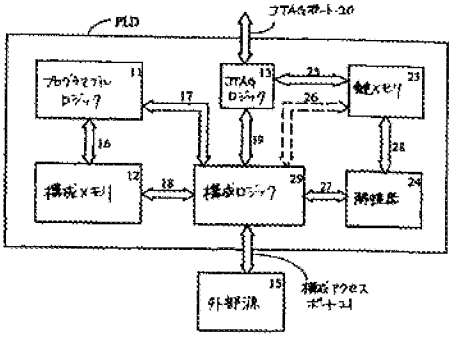


Fig. 3

【図４ａ】

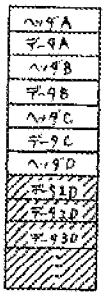


Fig. 4a

【図４ｂ】

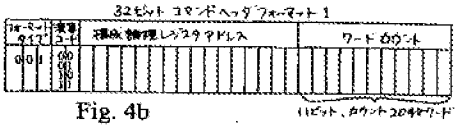


Fig. 4b

【図４ｃ】

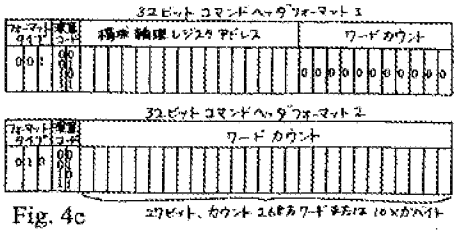
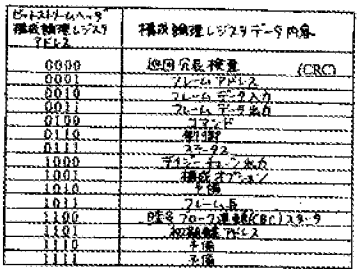


Fig. 4c

【図４ｄ】



ビットストリームヘッダ構成物理レジスタアドレス	構成物理レジスタデータ内容
0000	巡回冗長検査 (CRC)
0001	フレームアドレス
0010	フレームデータ入力
0011	フレームデータ出力
0100	コマンド
0110	制御
0111	ステータス
1000	ブルートフォース
1001	構成オプション
1010	予備
1011	フレームＢ
1100	時刻カウンタ(CRC)ステータス
1101	初期構成アドレス
1110	予備
1111	予備

Fig. 4d

【図５ａ】

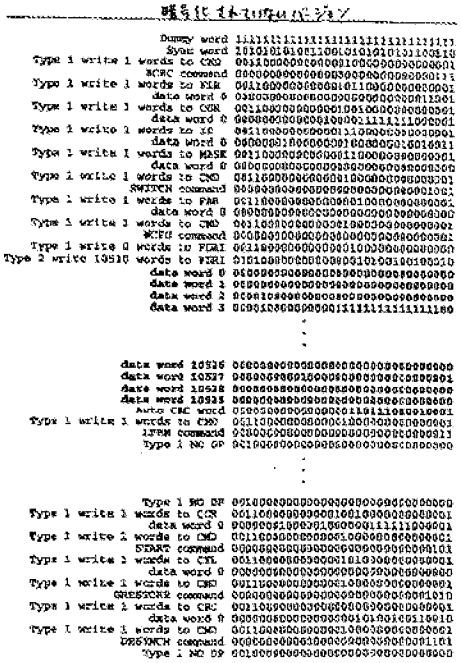


Fig. 5a

【図５ｂ】

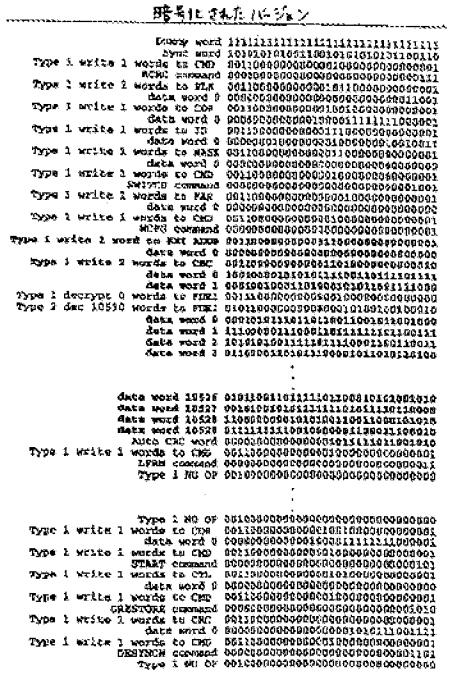


Fig. 5b

【図６】

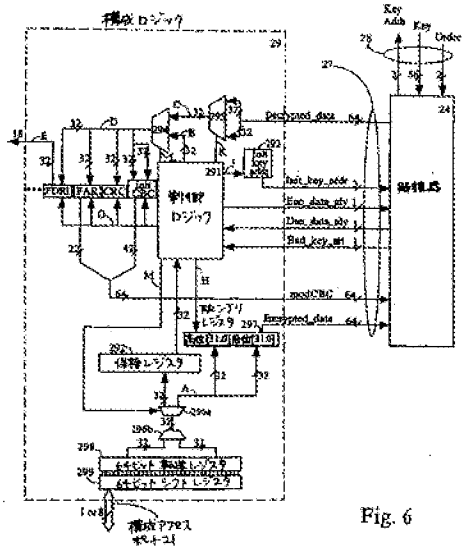


Fig. 6

【図７ａ】

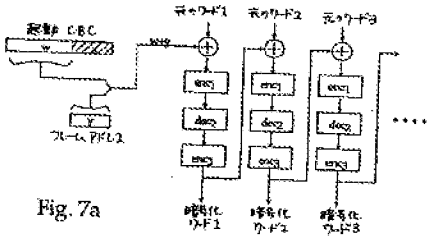


Fig. 7a

【図７ｂ】

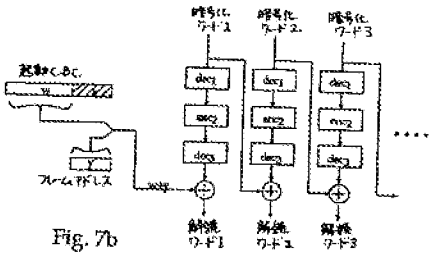


Fig. 7b

【図８】

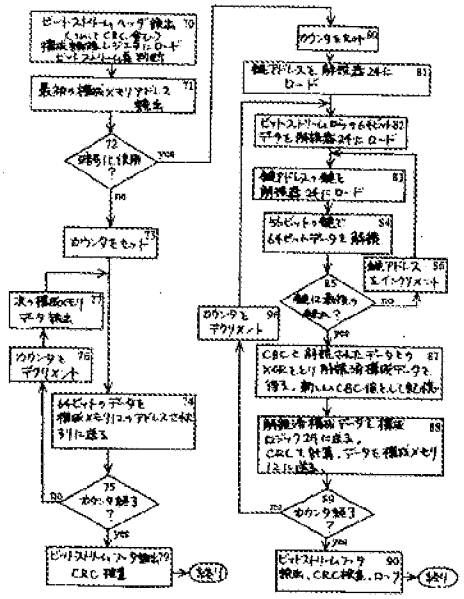


Fig. 8

【図９】

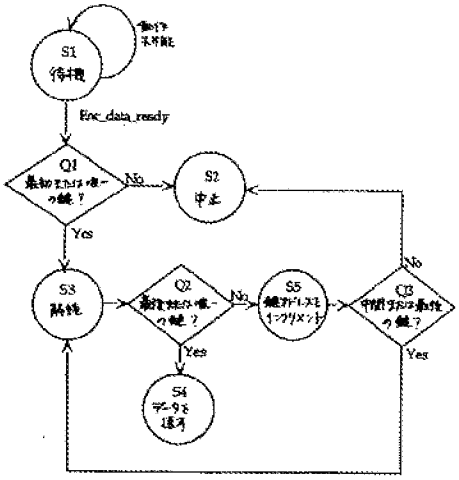


Fig. 9

【図１０ａ】

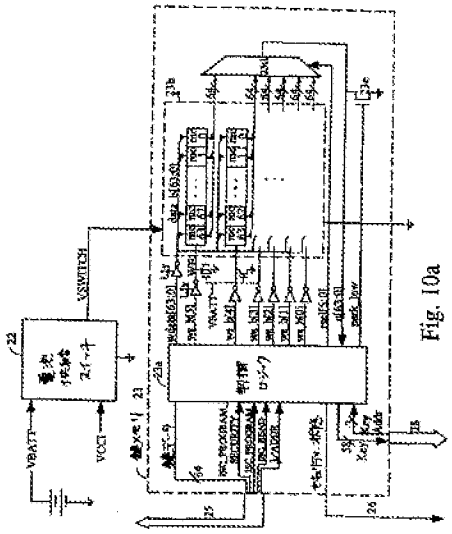
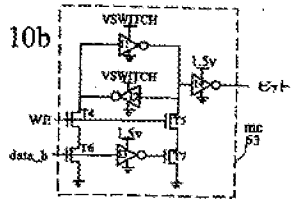


Fig. 10a

【図１０ｂ】

Fig. 10b



【図１１】

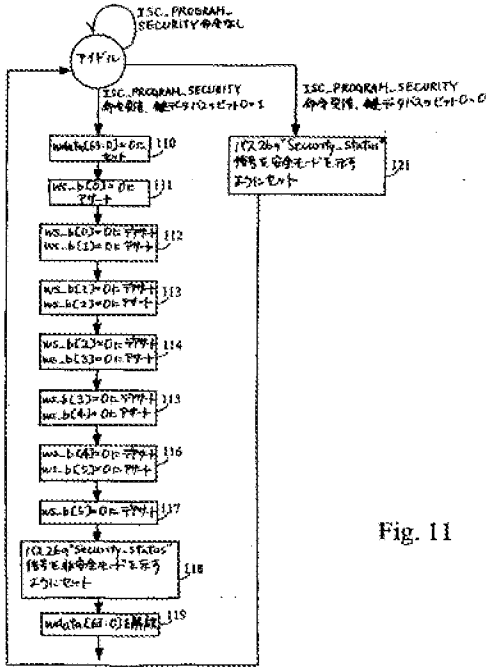


Fig. 11

【図１２】

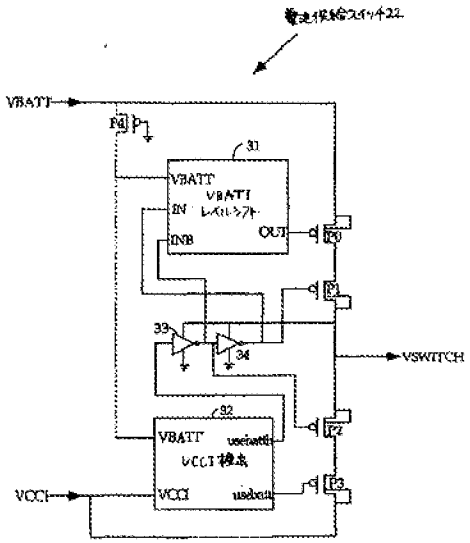


Fig. 12

【図１３】

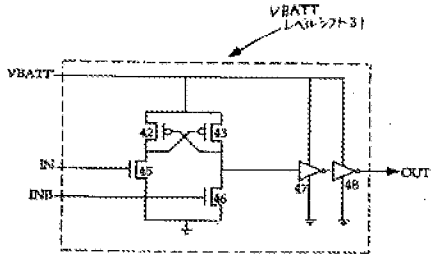


Fig. 13

【図１４】

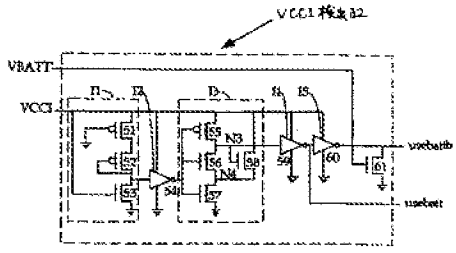


Fig. 14

【図１５】

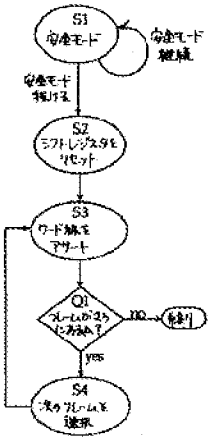


Fig. 15

【図１６】

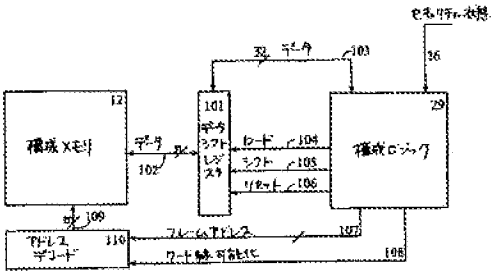


Fig. 16

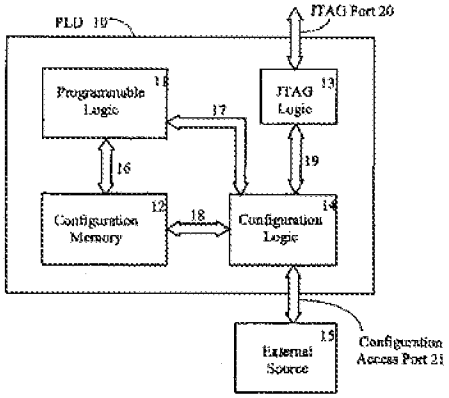


Fig. 1
PRIOR ART

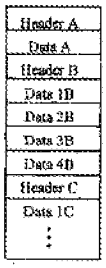


Fig. 2a
PRIOR ART

Bitstream header Config. Logic Reg. address	Config. Logic Register data contents
0000	Cyclic Redundancy Check (CRC)
0001	Frame Address
0010	Frame Data Input
0011	Frame Data Output
0100	Command
0110	Control
0111	Status
1000	Daisy Chain Output
1001	Configuration Option
1010	Reserved
1011	Frame Length
1100	Reserved
1101	Reserved
1110	Reserved
1111	Reserved

Fig. 2d
PRIOR ART

32-bit Command Header Format 1

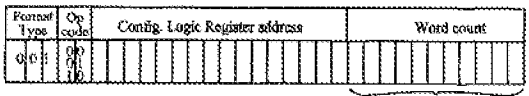
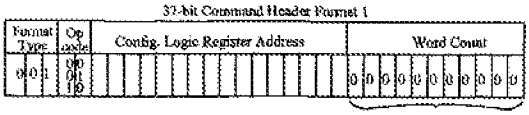


Fig. 2b PRIOR ART

32-bit Command Header Format 1



32-bit Command Header Format 2

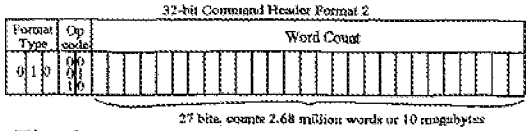


Fig. 2c PRIOR ART

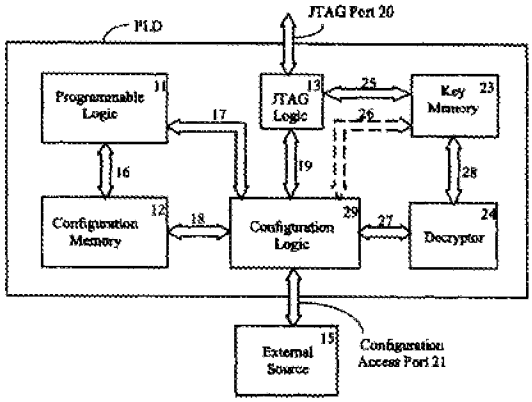

PLD
JTAG Port 20
Programmable Logic
11
JTAG Logic
13
Key Memory
23
25
26
17
16
19
28
Configuration Memory
12
18
Configuration Logic
29
27
Decryptor
24
External Source
15
Configuration Access Port 21

Fig. 3

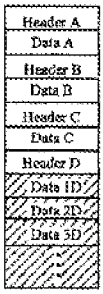


Fig. 4a

Bitstream header Config. Logic Reg. address	Config. Logic Register data contents
0000	Cyclic Redundancy Check (CRC)
0001	Frame Address
0010	Frame Data Input
0011	Frame Data Output
0100	Command
0110	Control
0111	Status
1000	Daisy Chain Output
1001	Configuration Option
1010	Reserved
1011	Frame Length
1100	Cipher block chaining(CBC) starter
1101	Initial key address
1110	Reserved
1111	Reserved

Fig. 4d

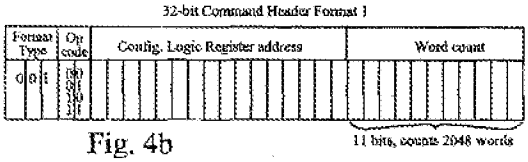


Fig. 4b

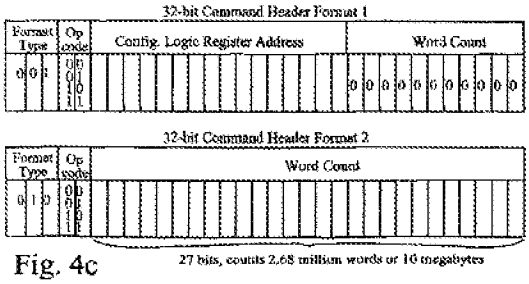


Fig. 4c

UNENCRYPTED VERSION

```
                    Dummy word 1111111111111111111111111111111111111111
                     Sync word 1010101010011001010101010101011001100110
     Type 1 write 1 words to CMD  0011000000000000000100000000000100000001
                    HCRC command 0000000000000000000000000000000000000111
     Type 1 write 1 words to FLR  0011000000000000010110000000000000000001
                    data word 0 0000000000000000000000000000000000001001
     Type 1 write 1 words to COR  0011000000000000100100000000000000000001
                    data word 0 0000000000000091000011111121000000001
     Type 1 write 1 words to ID   0011000000000000111000000000000000000001
                    data word 0 0000000100000000010000000000100000010011
     Type 1 write 1 words to MASK 0011000000000000111000000000000000000001
                    data word 0 0000000000000000000000000000000000000000
     Type 1 write 1 words to CMD  0011000000000000120010000000000000000001
                  SWITCH command 0000000100000000000000000000000100000001
     Type 1 write 1 words to FAR  0011000000000000100100000000000000000001
                    data word 0 0000000000000000000000000000000000000000
     Type 1 write 1 words to CMD  0011000000000000120010000000000000000001
                    MCFG command 0000000000000000000000000000000000000001
     Type 1 write 0 words to FDRI 0011000000000000100000000000000000000000
Type 2 write 10538 words to FDRI 0100000000000000010301001000190018
                    data word 0 0000000000000000000000000000000000000000
                    data word 1 0000000000000000000000000000000000000000
                    data word 2 0000000000000000000000000000000000000000
                    data word 3 0000100000000001111111111111111111111111

                                          .
                                          .
                                          .

                  data word 10526 0000000000000000000000000000000000000000
                  data word 10527 0000000000100000000000000000000000000001
                  data word 10528 0000000000000000000000000000000000000000
                  data word 10529 0000000000000000000000000000000000000000
                    Auto CRC word 0000000000000000001101110000100000000001
     Type 1 write 1 words to CMD  0011000000000000130000000000000000000001
                    LFRM command 0000000000000000000000000000000000000011
                    Type 1 NO OP 0010000000000000000000000000000000000000

                                          .
                                          .
                                          .

                    Type 1 NO OP 0010000000000000000000000000000000000000
     Type 1 write 1 words to COR  0011000000000000100100000000000000000001
                    data word 0 0000000100000000010000000000100000010011
     Type 1 write 1 words to CMD  0011000000000000120010000000000000000001
                    START command 0000000000000000000000000000000100000101
     Type 1 write 1 words to CTL  0011000000000000100100000000000000000001
                    data word 0 0000000000000000000000000000000000000000
     Type 1 write 1 words to CBC  0011000000000000100100000000000000000001
                  GRESTORE command 0000000000000000000000000000000000010110
     Type 1 write 1 words to CRC  0011000000000000100100000000000000000001
                    data word 0 0000000000000000000000000000000000000000
     Type 1 write 1 words to CMD  0011000000000000120010000000000000000001
                  DESYNCH command 0000000000000000000000000000000000001101
                    Type 1 NO OP 0010000000000000000000000000000000000000
```

Fig. 5a

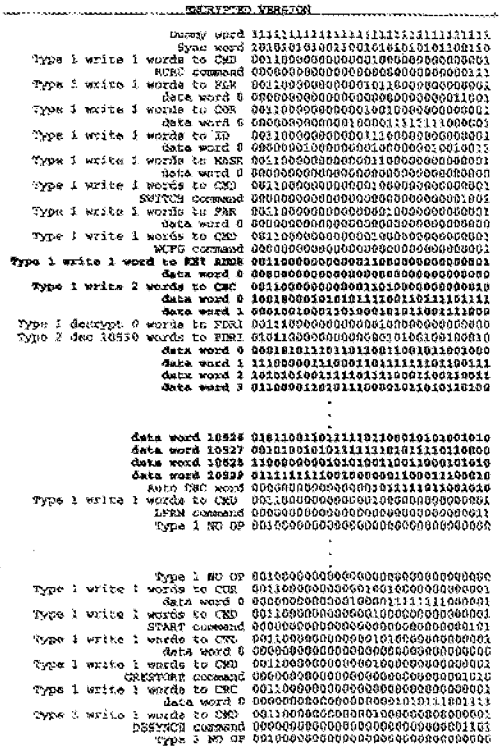

ENCRYPTED VERSION
Dummy word
Sync word
Type 1 write 1 words to CMD
RCRC command
Type 1 write 1 words to FAR
data word 0
Type 1 write 1 words to COR
data word 0
Type 1 write 1 words to IDR
data word 0
Type 1 write 1 words to MASK
data word 0
Type 1 write 1 words to CMD
SWITCH command
Type 1 write 1 words to FAR
data word 0
Type 1 write 1 words to CMD
WCFG command
Type 1 write 1 word to KEY ADDR
data word 0
Type 1 write 2 words to CBC
data word 0
data word 1
Type 1 decrypt 0 words to FDRI
Type 2 decrypt 10550 words to FDRI
data word 0
data word 1
data word 2
data word 3
data word 10525
data word 10526
data word 10527
Auto CRC word
Type 1 write 1 words to CMD
LFRM command
Type 1 NO OP
Type 1 NO OP
Type 1 write 1 words to COR
data word 0
Type 1 write 1 words to CMD
START command
Type 1 write 1 words to CTL
data word 0
Type 1 write 1 words to CMD
GRESTORE command
Type 1 write 1 words to CBC
data word 0
Type 1 write 1 words to CMD
DESYNCH command
Type 1 NO OP

Fig. 5b

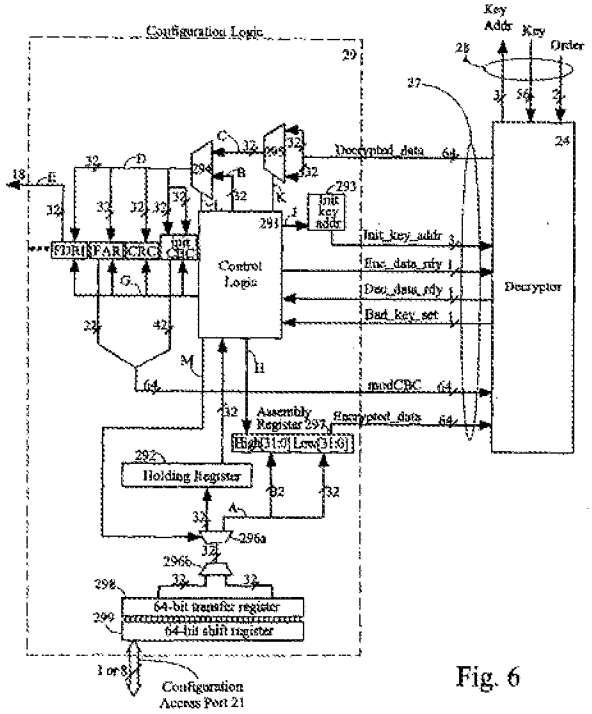

Configuration Logic
29
Key Addr
Key
Order
28
27
24
Decrypted_data
64
18
E
32
D
C
B
K
L
293
294
Init key addr
291
Init_key_addr
Enc_data_rdy
Dec_data_rdy
Bad_key_set
Control Logic
Decryptor
SYNC
FAR
CRC
Init CBC
G
42
M
H
modCBC
Assembly Register 297
Encrypted_data
High[31:0]
Low[31:0]
292
Holding Register
A
296a
2966
298
64-bit transfer register
299
64-bit shift register
1 or 8
Configuration Access Port 21

Fig. 6

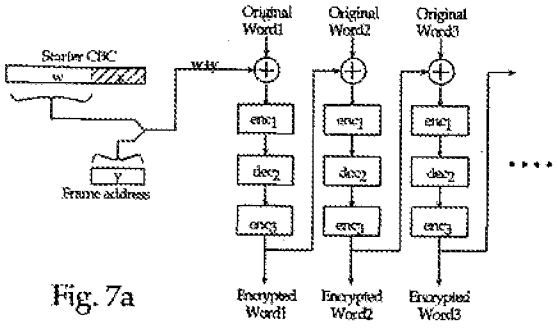

Original Word1
Original Word2
Original Word3
Starter CBC
w
w||x
enc1
dec2
enc3
y
Frame address
Encrypted Word1
Encrypted Word2
Encrypted Word3

Fig. 7a

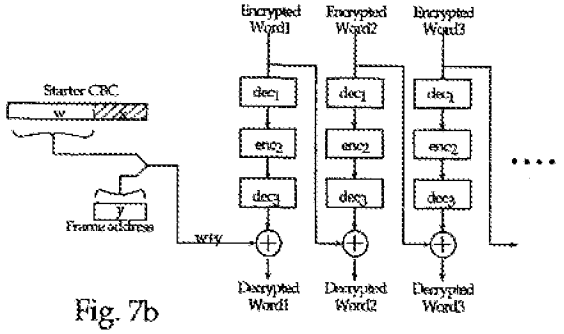

Encrypted Word1
Encrypted Word2
Encrypted Word3
Starter CBC
w
w||x
dec1
enc2
dec3
y
Frame address
w||y
Decrypted Word1
Decrypted Word2
Decrypted Word3

Fig. 7b

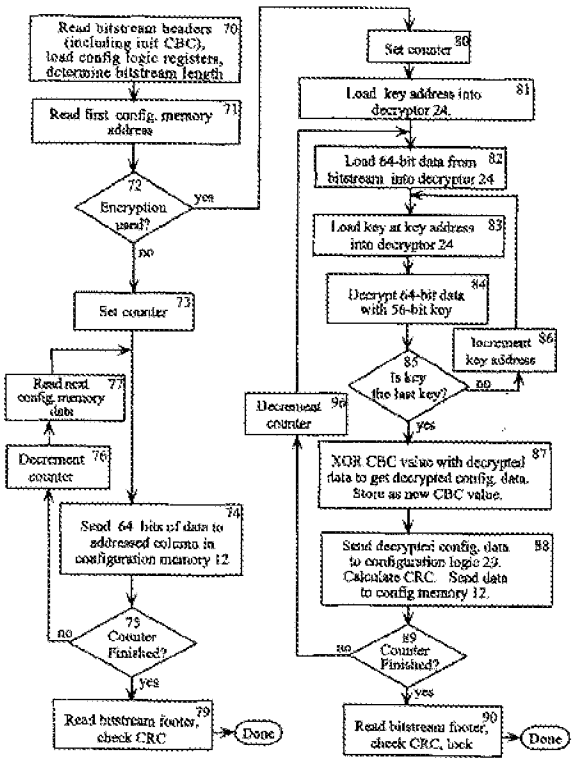

Read bitstream headers (including init CBC), load config logic registers, determine bitstream length 70
Read first config. memory address 71
Encryption used? 72
yes
no
Set counter 73
Read next config memory data 77
Decrement counter 76
Send 64 bits of data to addressed column in configuration memory 12 74
Counter Finished? 75
no
yes
Read bitstream footer, check CRC 79
Done
Set counter 80
Load key address into decryptor 24 81
Load 64-bit data from bitstream into decryptor 24 82
Load key at key address into decryptor 24 83
Decrypt 64-bit data with 56-bit key 84
Increment key address 86
Is key the last key? 85
no
yes
Decrement counter 96
XOR CBC value with decrypted data to get decrypted config. data. Store as new CBC value. 87
Send decrypted config. data to configuration logic 23. Calculate CRC. Send data to config memory 12. 88
Counter Finished? 89
no
yes
Read bitstream footer, check CRC, lock 90
Done

Fig. 8

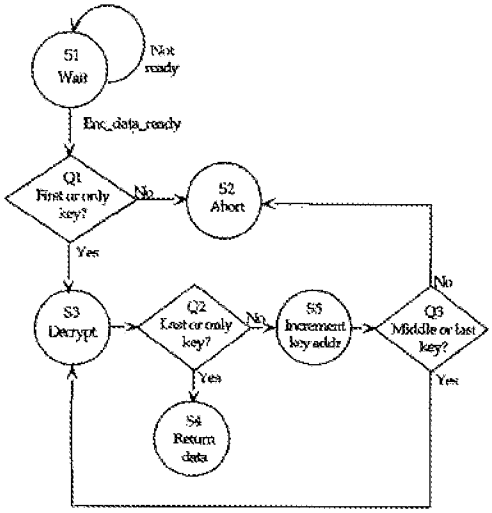

S1
Wait
Not
ready
Enc_data_ready
Q1
First or only
key?
No
S2
Abort
Yes
S3
Decrypt
Q2
Last or only
key?
No
S5
Increment
key addr
Q3
Middle or last
key?
No
Yes
Yes
S4
Return
data

Fig. 9

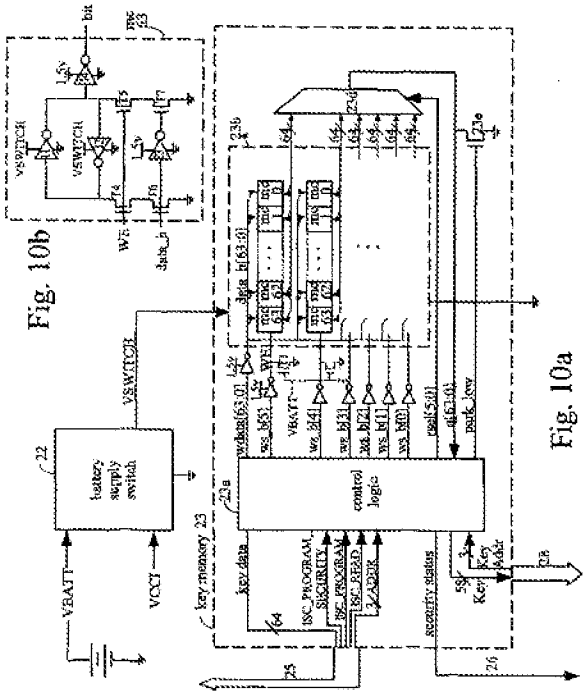


Fig. 10a

Fig. 10b

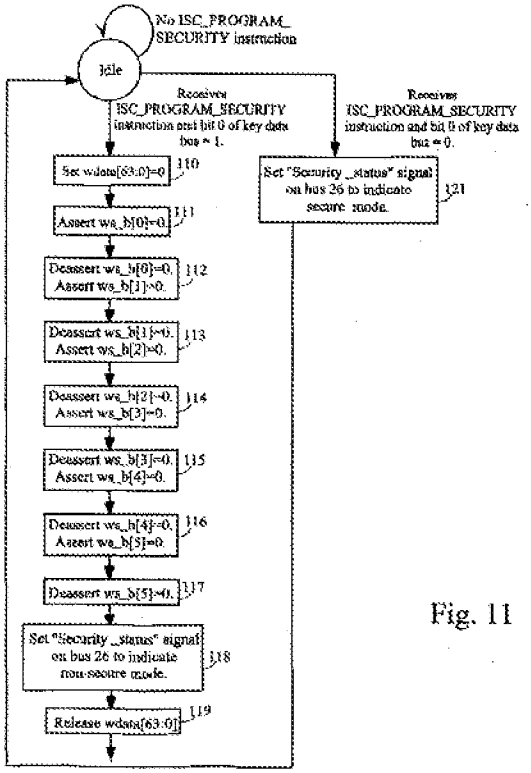

No ISC_PROGRAM_SECURITY instruction
Idle
Receives ISC_PROGRAM_SECURITY instruction and bit 0 of key data bus = 1.
Receives ISC_PROGRAM_SECURITY instruction and bit 0 of key data bus = 0.
Set wdata[63:0]=0
110
Assert ws_b[0]=0.
111
Deassert ws_b[0]=0.
Assert ws_b[1]=0.
112
Deassert ws_b[1]=0.
Assert ws_b[2]=0.
113
Deassert ws_b[2]=0.
Assert ws_b[3]=0.
114
Deassert ws_b[3]=0.
Assert ws_b[4]=0.
115
Deassert ws_b[4]=0.
Assert ws_b[5]=0.
116
Deassert ws_b[5]=0.
117
Set "Security_status" signal on bus 26 to indicate non-secure mode.
118
Release wdata[63:0]
119
Set "Security_status" signal on bus 26 to indicate secure mode.
121

Fig. 11

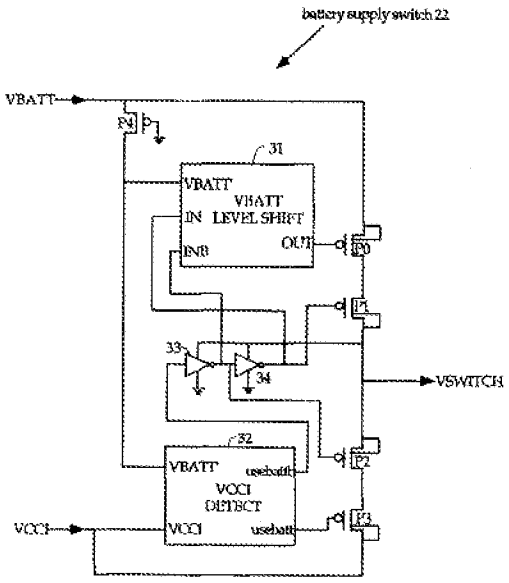

battery supply switch 22
VBATT
P4
31
VBATT
VBATT
IN LEVEL SHIFT
OUT
INB
P0
P1
33
34
VSWITCH
32
VBATT
usebatt
VCCI
DETECT
VCCI
usebatt
P2
P3
VCCI

Fig. 12

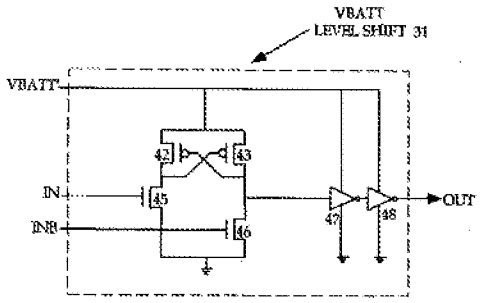

VBATT
LEVEL SHIFT 31
VBATT
42
43
IN
45
INB
46
47
48
OUT

Fig. 13

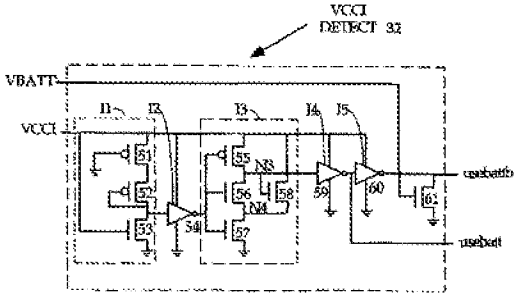

VCCI
DETECT 32
VBATT
I1
I2
I3
I4
I5
VCCI
51
52
53
54
55
56
57
N3
N4
58
59
60
61
usebatt

Fig. 14

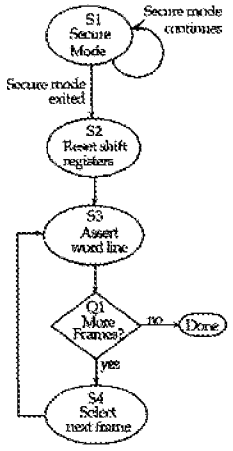

S1
Secure
Mode
Secure mode
continues
Secure mode
exited
S2
Reset shift
registers
S3
Assert
word line
Q1
More
Frames?
no
Done
yes
S4
Select
next frame

Fig. 15

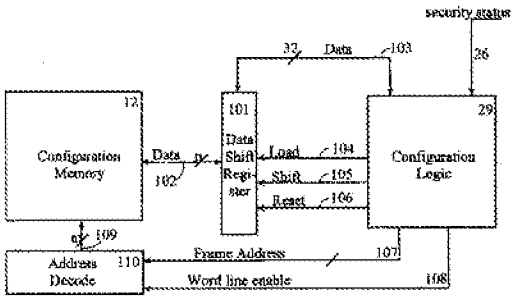

security status
32
Data
103
26
12
101
29
Configuration Memory
Data
n
102
Data Shift Register
Load
104
Shift
105
Reset
106
Configuration Logic
n
109
Address Decode
110
Frame Address
107
Word line enable
108

Fig. 16

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/45056

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 970 142 A (ERICKSON CHARLES R) 19 October 1999 (1999-10-19) abstract; figure 1 column 1, line 16 - line 32 column 1, line 59 -column 2, line 5	3,4
Y		1,5,6
Y	US 5 615 263 A (TAKAHASHI RICHARD J) 25 March 1997 (1997-03-25) abstract	1
P,Y	US 2001/015919 A1 (KEAR THOMAS A) 23 August 2001 (2001-08-23) abstract; figure 5 page 9, paragraph 128 -page 10, paragraph 135 page 1, paragraph 9 -page 2, paragraph 11	5,6

-/--

[X] Further documents are listed in the continuation of box C. [X] Patent family members are listed in annex.

* Special categories of cited documents :
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 June 2003	27/06/2003
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Powell, D

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/45056

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 321 704 A (ERICKSON CHARLES R ET AL) 14 June 1994 (1994-06-14) cited in the application claims 1-3	4

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/45656

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
US 5970142	A	19-10-1999	US	6212639	B1	03-04-2001
US 5615263	A	25-03-1997	NONE			
US 2001015919	A1	23-08-2001	EP	1124330	A2	16-08-2001
			US	20010374458	A1	01-11-2001
			AU	2269301	A	03-07-2001
			WO	0146819	A1	28-06-2001
			GB	2375418	A	13-11-2002
US 5321704	A	14-06-1994	CA	2058684	A1	17-07-1992
			EP	0495642	A2	22-07-1992
			JP	8008758	A	12-01-1996
			US	5598424	A	28-01-1997

フロントページの続き

(74)代理人 100098316
弁理士 野田 久登
(74)代理人 100109162
弁理士 酒井 將行
(72)発明者 パン，レイモンド・シー
アメリカ合衆国、９５１２０ カリフォルニア州、サン・ノゼ、ファルコン・リッジ・コート、１１３８
(72)発明者 シー，ウォルター・エヌ
アメリカ合衆国、９４０２４ カリフォルニア州、ロス・アルトス・ヒルズ、コリーン・ドライブ、１９４８
(72)発明者 ウォン，ジェニファー
アメリカ合衆国、９４５３９ カリフォルニア州、フリーモント、エンカント・ウェイ、４０５６５
(72)発明者 トリンバーガー，スティーブン・エム
アメリカ合衆国、９５１２０ カリフォルニア州、サン・ノゼ、シャトー・ドライブ、１２６１
(72)発明者 センディーン，ジョン・エム
アメリカ合衆国、９４７０９ カリフォルニア州、バークリー、マーティン・ルーサー・キング・ジュニア・ウェイ、１４３５、ナンバー・５
(72)発明者 ラオ，カメスワラ・ケイ
アメリカ合衆国、９５１２９ カリフォルニア州、サン・ノゼ、グレイウッド・ドライブ、１４０２

Ｆターム(参考) 5B017 AA03 BA07 CA11
5J042 BA11 CA13 CA20 CA26 DA06
5J104 AA12 AA16 AA32 AA44 AA45 EA04 EA20 NA02 NA27 NA42
PA14

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成17年12月22日(2005.12.22)

【公表番号】特表2004-515001(P2004-515001A)
【公表日】平成16年5月20日(2004.5.20)
【年通号数】公開・登録公報2004-019
【出願番号】特願2002-546976(P2002-546976)
【国際特許分類第７版】
　Ｇ０６Ｆ　12/14
　Ｈ０３Ｋ　19/173
　Ｈ０４Ｌ　 9/10
【ＦＩ】
　Ｇ０６Ｆ　12/14　　３２０Ｂ
　Ｈ０３Ｋ　19/173　　１０１
　Ｈ０４Ｌ　 9/00　　６２１Ａ

【手続補正書】
【提出日】平成16年11月8日(2004.11.8)
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正の内容】
【特許請求の範囲】
　【請求項１】　構成メモリにより構成される構成可能ロジックと、
　ＰＬＤの外部源からビットストリームを受けるための構造とを含み、
　ビットストリームは、暗号化されていない構成ビットおよび暗号化された構成ビットを含み、さらに、
　解読鍵を記憶するための鍵メモリと、
　鍵を用いてビットストリーム中の暗号化された構成ビットを解読することにより構成データを形成するための解読アルゴリズムを有する解読器と、
　構成データを構成メモリにロードするための構造とを含む、プログラマブルロジックデバイス（ＰＬＤ）。
　【請求項２】　ビットストリームが暗号化されたデータを含むかどうかを示すヘッダ情報をビットストリームから読出すための構造と、
　ビットストリームが暗号化されたデータを含むことをヘッダ情報が示す場合にはビットストリームを解読器に方向付け、ビットストリームが暗号化されたデータを含まないことをヘッダ情報が示す場合には解読器を飛び越すための構造とをさらに含む、請求項１に記載のＰＬＤ。
　【請求項３】　構成メモリから構成をリードバックするための構造と、
　ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すとき構成をリードバックするための構造を不能化するための構造とをさらに含む、請求項２に記載のＰＬＤ。
　【請求項４】　ＰＬＤが構成された後ＰＬＤを再構成するための構造と、
　ビットストリームが暗号化されたデータを含むことをヘッダ情報が示すときＰＬＤを再構成するための構造を不能化するための構造とをさらに含む、請求項２に記載のＰＬＤ。
　【請求項５】　鍵メモリは、複数個の解読鍵を記憶するための複数個のレジスタを含み、
　解読器は、複数個の解読鍵のうち少なくとも１つを用いてビットストリーム内のデータ

を解読するための解読アルゴリズムを有する、請求項１に記載のＰＬＤ。

【請求項６】　解読器は、複数個の解読鍵を記憶するためのレジスタの１つから、解読のために別の鍵も使用されるかどうかを示す値を読出す、請求項５に記載のＰＬＤ。

【請求項７】　鍵は、当該鍵が、鍵の組のうちの最後の鍵であるか、または最後の鍵でないかを特定する、請求項５に記載のＰＬＤ。

【請求項８】　ビットストリーム内の第１の群のワードは、第１の設計者に既知の第１の鍵によって暗号化され、ビットストリーム内の第２の群のワードは、第２の設計者に既知の第２の鍵によって暗号化される、請求項５に記載のＰＬＤ。

【請求項９】　鍵メモリを安全モードおよび非安全モードにするための構造をさらに含み、鍵メモリが非安全モードにある間に解読鍵がロードされる、請求項１に記載のＰＬＤ。

【請求項１０】　安全モードから非安全モードに鍵メモリを移すことにより、解読鍵および構成データの少なくとも１つが消去される、請求項９に記載のＰＬＤ。